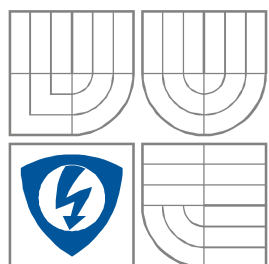




VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH
TECHNOLOGIÍ
ÚSTAV BIOMEDICÍNSKÉHO INŽENÝRSTVÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF BIOMEDICAL ENGINEERING

KOMPRESSE SIGNÁLŮ EKG

COMPRESSION OF ECG SIGNALS

BAKALÁŘSKÁ PRÁCE

BACHELOR'S PROJECT

AUTOR PRÁCE

AUTHOR

Jan Kašpárek

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. Jan Hrubeš

BRNO, 2011



VYSOKÉ UČENÍ
TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

Ústav biomedicínského inženýrství

Bakalářská práce

bakalářský studijní obor
Biomedicínská technika a bioinformatika

Student: Jan Kašpárek
Ročník: 3

ID: 119718
Akademický rok: 2010/2011

NÁZEV TÉMATU:

Kompresce signálů EKG

POKYNY PRO VYPRACOVÁNÍ:

Seznamte se s bezztrátovými kompresními metodami RLE, LZ a Huffmanovým kódováním. Implementujte kompresní a dekompresní algoritmus RLE pro kompresi signálu EKG a otestujte. Rozšiřte RLE kodér o Burrows-Wheelerovu transformaci a otestujte na knihovně signálů EKG. Srovnajte výsledky obou variant s jinými bezztrátovými kodéry.

DOPORUČENÁ LITERATURA:

- [1] SALOMON, D. Data Compression - The Complete Reference, 4/E. London: Springer, 2007.
- [2] DAKUA S.P., SAHAMBI, J.S. Lossless ECG Compression for Event Recorder Based on Burrows-Wheeler Transformation and Move-To-Front Coder. International Journal of Recent Trends in Engineering, Vol 1, No. 3, May 2009.

Termín zadání: 7.2.2011

Termín odevzdání: 27.5.2011

Vedoucí práce: Ing. Jan Hrubeš

prof. Ing. Ivo Provazník, Ph.D.
Předseda oborové rady

UPOZORNĚNÍ:

Autor bakalářské práce nesmí při vytváření bakalářské práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

Abstrakt

V této práci je obecně popsáno několik základních kompresních algoritmů využívaných jak při ztrátové, tak při neztrátové kompresi. Jmenovitě se jedná o proudové kódování, Lempel-Ziv, Huffmanovo kódování, Move-to-Front, vlnkovou transformaci a Burrows-Wheelerovu transformaci.

Cílem práce je vytvořit kompresní algoritmus EKG s implementací proudového kódování a BWT. Dalším postupem použitým v algoritmu je vlnková transformace. Algoritmus je napsán v programovém prostředí MATLAB.

Klíčová slova

Komprese, EKG, RLE, BWT, DWT, adaptivní, práh.

Abstract

This thesis contains general description of several basic compression algorithms used either for lossy and lossless compression. These algorithms are runlength encoding, Lempel-Ziv, Huffman encoding, Move-to-Front, wavelet transform and Burrows-Wheeler transform.

Aim of this thesis is creation of ECG compression algorithm with implementation of runlength encoding and BWT. Another technique used in algorithm is wavelet transform. The algorithm is written in programming environment MATLAB.

Key words

Compression, ECG, RLE, BWT, DWT, adaptive, threshold.

KAŠPÁREK, J. *Komprese signálů EKG*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2011. 42 s. Vedoucí bakalářské práce: Ing. Jan Hruběš.

Prohlášení

Prohlašuji, že svou bakalářskou práci na Komprese EKG jsem vypracoval samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené bakalářské práce dále prohlašuji, že v souvislosti s vytvořením této práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení § 152 trestního zákona č. 140/1961 Sb.

V Brně dne

.....

podpis autora

Poděkování

Děkuji vedoucímu bakalářské práce Ing. Janu Hruběšovi za účinnou metodickou, pedagogickou a odbornou pomoc a další cenné rady při zpracování mé bakalářské práce.

V Brně dne

.....

podpis autora

Obsah

Úvod	1
1 Komprese	2
1.1 Bezztrátová komprese	3
1.2 Ztrátová komprese	4
2 Postupy komprese	5
2.1 Proudové kódování	5
2.1.1 Příklad proudového kódování	5
2.2 Lempel-Ziv	6
2.2.1 Příklad komprese Lempel-Ziv	6
2.3 Huffmanovo kódování	7
2.3.1 Příklad Huffmanova kódování	9
2.4 Move-to-Front kódování	9
2.4.1 Příklad MTF kódování	10
2.5 Vlnková transformace	11
2.5.1 Spojitá vlnková transformace	12
2.5.2 Diskrétní vlnková transformace	12
2.6 Burrows-Wheelerova transformace	14
2.6.1 Příklad Burrows-Wheelerovy transformace	15
2.6.2 Příklad inverzní Burrows-Wheelerovy transformace	15
3 EKG	17
3.1 Objem snímaných dat	17
4 Implementace proudového kódování pro EKG	19
4.1 Zhodnocení výsledků	20
5 Implementace vlnkové transformace	22
5.1 Modifikace proudového kódování	23
5.2 Parametry vlnkové transformace	23
5.2.1 Vlnka	23
5.2.2 Počet pásem transformace	24
5.2.3 Práh	25
5.3 Zhodnocení výsledků	25
5.3.1 Stanovení zkreslení signálu	25
5.3.2 Vliv velikosti prahu	26

5.3.3	Vliv počtu pásem.....	27
5.3.4	Výsledný signál	29
6	Implementace Burrows-Wheelerovy transformace.....	33
6.1	Aplikace BWT přímo na EKG signál	33
6.2	Aplikace BWT na koeficienty DWT	33
6.2.1	Zhodnocení výsledků	35
7	Srovnání algoritmů.....	37
	Závěr.....	39
	Seznam použité literatury	41
	Seznam použitých zkratek.....	42

Seznam obrázků

Obrázek 1. Možné podoby Huffmanových stromů [3]	7
Obrázek 2. Příklad Huffmanova kódování. Převzato z [5] a upraveno.....	8
Obrázek 3. Příklad MTF kódování.....	10
Obrázek 4. Vlnková reprezentace EKG signálu.....	11
Obrázek 5. Znázornění frekvenčního rozkladu a uspořádání rozkladových filtrů [9]	13
Obrázek 6. Znázornění rozlišení v jednotlivých pásmech [8].....	13
Obrázek 7. Příklad vizualizace koeficientů diskrétní vlnkové transformace	14
Obrázek 8. Příklad Burrows-Wheelerovy transformace	15
Obrázek 9. Příklad inverzní Burrows-Wheelerovy transformace	16
Obrázek 10. Průběh jedné periody idealizovaného EKG. Převzato z [12].	17
Obrázek 11. Ukázka doplnění svodů nulami	19
Obrázek 12. Průběh reálného záznamu EKG.....	20
Obrázek 13. Diagram procesu komprese signálu.....	22
Obrázek 14. Diagram procesu dekomprese signálu	22
Obrázek 15. Dekompoziční funkce vlnky bior3.1	23
Obrázek 16. Rekonstrukční funkce vlnky bior3.1.....	23
Obrázek 17. Hodnoty CR a PRD pro různé vlnky	24
Obrázek 18. Závislost CR a PRD na hodnotě prahu	26
Obrázek 19. Závislost CR a PRD na počtu pásem transformace při prahu $0,19 \sigma$	28
Obrázek 20. Závislost CR a PRD na počtu pásem transformace při prahu $0,3 \sigma$	28
Obrázek 21. Původní signál prvního svodu ze souboru MO1_125_03.....	30
Obrázek 22. Stejný signál po rekonstrukci.....	30
Obrázek 23. Rozdílový signál původního a rekonstruovaného průběhu ($10\times$ zvětšeno)	31
Obrázek 24. Detail původního signálu.....	31
Obrázek 25. Detail rekonstruovaného signálu	32
Obrázek 26. Detail rozdílového signálu ($10\times$ zvětšeno)	32
Obrázek 27. Diagram procesu komprese	33
Obrázek 28. Diagram procesu dekomprese.....	33
Obrázek 29. Průběh koeficientů posledního pásma vlnkové transformace	34
Obrázek 30. Průběh koeficientů posledního pásma DWT po aplikaci BWT.....	34
Obrázek 31. Rozdíl CR při aplikaci BWT oproti kompresi bez BWT	35

Seznam tabulek

Tabulka 1. Hodnoty CR a PRD pro jmenovité hodnoty prahu	27
Tabulka 2. Hodnoty CR a PRD pro různé počty pásem při jmenovitých hodnotách prahu	29
Tabulka 3. Rozdíl CR při aplikaci BWT oproti kompresi bez BWT	36
Tabulka 4. Jednoznačné označení algoritmů.....	37
Tabulka 5. Porovnání s jinými algoritmy	37

Úvod

Snímání EKG představuje činnost vyúsťující v ukládání stále rostoucího proudu dat. Objem dat je závislý jak na délce snímání, tak na požadované kvalitě snímaného signálu. Ne vždy se setkáváme s vysokými požadavky na kvalitu signálu, přesto lze zvláště u dlouhodobých záznamů hovořit o nezanedbatelných nárocích na kapacitu médií. Komprese EKG je procesem snižování těchto nároků.

Tato práce obsahuje popis principu několika základních metod komprese včetně těch, které jsou ve výsledném algoritmu použity. U popisu každé metody je uveden také názorný příklad jejího použití.

Cílem této práce je implementovat algoritmus proudového kódování a Burrows-Wheelerovy transformace pro efektivní kompresi EKG. Kromě těchto dvou postupů je použita i vlnková transformace. Veškerá praktická práce je prováděna v programovém prostředí MATLAB.

1 Komprese

Požadavky na kapacitu médií s postupem času stále rostou. Zvyšuje se nejen množství ukládaných dat, ale také jejich bitová hloubka. Například obrazové a zvukové stopy mají stále větší a větší datový tok. Nejzřetelnější je tato skutečnost právě u videa, kde můžeme sledovat postupný přechod z DVD na Blu-ray. Do budoucna se dá očekávat pouze další nárůst. Komprese dat (někdy se objevuje také pojem komprimace dat) je postup, jehož cílem je zmenšit výslednou velikost dat. Využití komprese se neomezuje jen na ukládání dat na fyzická média jako jsou pevné disky, popřípadě CD nebo DVD. Snad ještě markantnější využití nachází při přenosu dat přes počítačové sítě, kde můžeme kompresí výrazně zkrátit dobu přenosu.

Základním principem komprese je odstranění redundantních informací ze souboru dat. Velmi často se jedná o informace, které se v souboru opakují, nebo informace, které mohou být odvozeny ze zbytku.

K praktickému použití komprese je nutné využití dvou zpravidla samostatných algoritmů. Jeden algoritmus obstarává kompresi (často nazývaný kodér), zatímco druhý dekompresi (často označovaný jako dekodér). Přičemž oba tyto algoritmy mohou mít rozdílné nároky na výpočetní výkon, z čehož vyplývá i rozdílná doba komprese a dekomprese. Ta může být kromě kompresního poměru a zkreslení originálních dat rovněž kritériem hodnocení algoritmu.

V současnosti existuje velké množství kompresních algoritmů, současně existuje i několik hledisek, podle kterých je lze dělit:

Podle počtu průchodů souborem dat:

- Jednoprůchodové – algoritmus projde soubor dat pouze jednou a ihned komprimuje.
- Dvou(více)průchodové – algoritmus prochází soubor dat alespoň dvakrát. Při prvním průchodu nedochází ke kompresi, pouze se zjišťují parametry vstupních dat. Samotná komprese nastává až při druhém průchodu.

Podle symetrie komprese a dekomprese:

- Symetrické – procesy komprese i dekomprese jsou zhruba stejně náročné a pro jeden soubor dat se tedy doba jejich trvání nebude výrazně lišit.

- Asymetrické – komprese, nebo dekomprese je výrazně náročnější. Jedna z těchto operací bude trvat výrazně déle. Náročnější operací bývá zpravidla komprese.

A nakonec nejdůležitější dělení podle schopnosti rekonstruovat komprimovaná data:

- Ztrátové – část informace je kompresí ztracena, nebo zkreslena.
- Bezztrátové – po dekompresi dostáváme soubor dat identický k původnímu souboru.

1.1 Bezztrátová komprese

Bezztrátová komprese je proces komprese, při němž lze dosáhnout dokonalé rekonstrukce původních informací. Nedochází tedy k žádné ztrátě. Dekomprimovaná data jsou identická se vstupními. Tyto postupy komprese bývají využívány právě tam, kde neúplná informace není dostačující. Například komprese textu. Cenou za dokonalé zachování informací je relativně nízký kompresní poměr, řádově můžeme počítat s hodnotami kolem 2:1 [1]. Limit kompresního poměru zde podle [2] představuje entropie zpracovávaného souboru dat.

Postupy bezztrátové komprese bývají pro svou spolehlivost a jednoduchost často součástí různých sofistikovanějších přístupů včetně ztrátové komprese.

Základní metody bezztrátové komprese lze podle přístupu rozdělit na několik druhů uvádím rozdělení podle [3]:

- Základní algoritmy – princip těchto algoritmů není jednotný, přesto se jedná o jednoduché, až intuitivní postupy. Řadí se sem například proudové kódování nebo Move-to-Front.
- Statistické algoritmy – pracují na základě četnosti symbolů. Například Huffmanův, nebo aritmetický kódér.
- Slovníkové algoritmy – nahrazují opakovaný výskyt symbolu odkazem na jeho předchozí pozici. Rodina LZ kódérů.
- Transformace vstupních dat – Vstupní data jsou před samotnou kompresí nejdříve transformována. Transformovaná data mají v ideálním případě lepší předpoklady pro následnou kompresi. Využívá se například Burrows-Wheelerova transformace.

1.2 Ztrátová komprese

Jak už z názvu vyplývá, ztrátová komprese je proces, při němž dochází ke ztrátě informací. Tato ztráta je ovšem kompenzována výrazně vyššími kompresními poměry, než u bezztrátové komprese. Při ztrátové kompresi je snahou odstranit nejen redundantní informace, ale i informace méně důležité. Kromě toho zpravidla dochází i k nechtěnému zkreslení. Díky výrazně lepším kompresním poměrům je ztrátová komprese mnohem rozšířenější, než bezztrátová. Je využívána zejména při kompresi obrazu nebo zvuku, kde se v úvahu bere i nedokonalost lidských smyslů a schopnost „domyslet si“.

Vstupní data bývají často nejdříve transformována a teprve potom jsou uplatněny samotné kompresní algoritmy. Příkladem může být využití vlnkové transformace a následné prahování jejích koeficientů. Postup dekomprese je přesně opačný, nejdříve se aplikují dekompresní algoritmy, poté inverzní transformace. Samotná ztráta informací může potom být způsobena jak samotným procesem transformace (často i jen zaokrouhlovací chyby různých koeficientů – takto vzniklé zkreslení je ale minimální), tak úmyslným potlačením některých informací.

2 Postupy komprese

Na jakoukoli metodu komprese lze nahlížet jako na transformaci vstupních dat, jejímž účelem je zmenšit jejich objem. Většina kompresních postupů toho při své aplikaci za předpokladu vhodné podoby vstupních dat dosahuje. Například proudové nebo Huffmanovo kódování.

U jiných postupů naopak dochází k expanzi dat nezávisle na podobě vstupu. Přesto jsou pro kompresi hojně využívány. Jejich cílem není přímo zmenšit velikost dat, ale transformovat data do podoby vhodnější pro postupy, které následně provedou skutečnou kompresi. Mezi tyto expanzní postupy v kompresi patří například vlnková nebo Burrows-Wheelerova transformace. Samy o sobě vlastně s kompresí mají společného jen málo, přesto si dovolím zařadit je zde mezi postupy kompresní.

2.1 Proudové kódování

Proudové kódování je pravděpodobně nejjednodušším způsobem komprese. Algoritmus je zaměřený na eliminaci opakujících se znaků. To probíhá tak, že každému znaku sekvence je přidělen další znak nesoucí hodnotu jeho opakování v řadě. Bezprostřední opakovaný výskyt původního znaku je před postupem k dalšímu prvku odstraněn.

2.1.1 Příklad proudového kódování

Vstupní sekvence => Výstupní sekvence
493333599999 => 1419431559

První znak výstupní sekvence značí počet opakování následujícího znaku. V této nejjednodušší formě proudového kódování dochází při vstupu sekvence neopakujících se znaků ke zvětšení objemu dat (expanzi), protože i těmto neopakujícím se znakům je přiřazován počet opakování (jednička). Tohoto neduhu se lze zbavit tím, že každá komprimovaná sekvence bude označena indikátorem komprese. Jako indikátor komprese musí být zvolen znak, který se v původní sekvenci nevyskytuje. V takovém případě by se původní příklad změnil na:

Vstupní sekvence => Výstupní sekvence
493333599999 => 49C435C59

V tomto demonstračním případě byl jako indikátor komprese zvolen znak „C“. Indikátor můžeme považovat za první znak komprese, druhý znak komprese potom nese

informaci o počtu opakování třetího znaku. Z toho vyplývá, že kódování znaku s menším počtem opakování, než čtyři, nepřinese žádnou úsporu. Kódují se proto jen znaky s větším počtem opakování. Způsob kódování s použitím indikátoru komprese bývá označován jako modifikované proudové kódování.

Z podstaty tohoto kódování vyplývá, že výsledný kompresní poměr je silně závislý na podobě vstupních dat. Vysoký kompresní poměr lze očekávat tam, kde se vyskytují dlouhé sekvence opakujeících se znaků. Naopak pro běžný text je proudové kódování samo o sobě prakticky nepoužitelné. Přesto se používá u faxů, kde je veškerý obsah zprávy kódován jako obrázek.

2.2 Lempel-Ziv

Algoritmus Lempel-Ziv (LZ) se poprvé objevil v roce 1977, kdy byl napsán pány Abrahamem Lempelem a Jacobem Zivem [4]. Tento algoritmus, podle roku vzniku označován jako LZ77, byl dále modifikován a s přispěním dalších tvůrců tak vznikly například algoritmy LZSS nebo LZX. O rok později vytvořili Lempel a Ziv druhou verzi svého algoritmu nazvanou LZ78. I tento algoritmus se dočkal mnoha pozdějších modifikací, například LZW. Je vidět, že rodina algoritmů LZ čítá hodně členů. Pro ilustraci jejich základní funkce se budu držet principu prvního z nich, tedy LZ77.

Jedná se o slovníkový algoritmus. Podobně jako proudové kódování hledá opakujeící se části dat. Rozdíl je v tom, že LZ77 se neomezuje jen na opakování jednotlivých znaků, které by spolu musely přímo sousedit. Cílem algoritmu je detekovat co nejdelší sekvence opakujeících se znaků napříč celým souborem dat. Jakmile algoritmus nalezne sekvenci opakujeící se v souboru dat, uloží na pozici jejího opakování odkaz skládající se z polohy dřívějšího výskytu a počtu shodných znaků. I zde je třeba odlišit odkazy od původních dat a stanovit minimální délku kódovaných sekvencí tak, aby jejich kódování přineslo úsporu místa.

2.2.1 Příklad komprese Lempel-Ziv

Vstupní řetězec => Výstupní řetězec

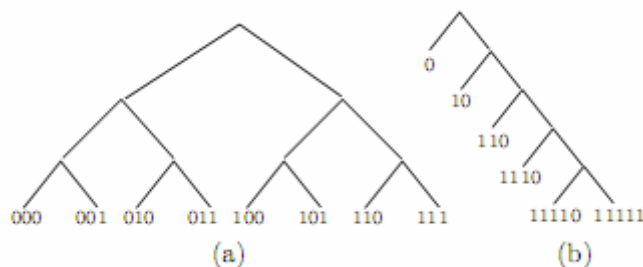
Leze leze po železe. => Leze l[2,4]po že[6,4].

V příkladu uvažuji jako znak i mezeru a rozlišuji velkou a malou abecedu. Odkazy jsou zapsány uvnitř hranatých závorek a skládají se ze dvou číselných údajů. První číslo určuje pozici prvního znaku původního výskytu opakujeící se sekvence, zatímco druhé číslo je délka opakujeící se sekvence, tedy počet znaků, které odkaz reprezentuje.

Algoritmus pracuje s využitím dvou oken, jedno okno slouží jako referenční, hledáme v něm tedy shodu s obsahem okna druhého. Velikost oken je přitom parametr určující vlastnosti komprese. S rostoucí velikostí referenčního okna je algoritmus schopen detekovat shodné sekvence o větší délce a tím pádem i dosáhnout lepšího kompresního poměru. Současně s tím ale rostou i jeho výpočetní požadavky. Většinou je rozhodnutí o délce okna přeneseno na uživatele, jemuž je dána možnost zvolit vlastnosti komprese podle vlastních potřeb.

2.3 Huffmanovo kódování

Huffmanovo kódování je zástupcem statistických kompresních algoritmů. Jako ostatní statistické postupy, i tento je dvouprůchodový. Tato myšlenka se objevila v roce 1952 a jejím autorem je David A. Huffman [4]. Algoritmus pracuje s pravděpodobností výskytu jednotlivých znaků (znakem zde může být chápáno i slovo) v souboru dat. Vytvoření tabulky pravděpodobností jednotlivých znaků je prvním krokem celého postupu a tvoří se při prvním průchodu souborem dat. Jakmile je tabulka vytvořena, zakóduje algoritmus jednotlivé znaky tak, aby znaky s největší četností (pravděpodobností výskytu) byly reprezentovány nejkratším kódem.

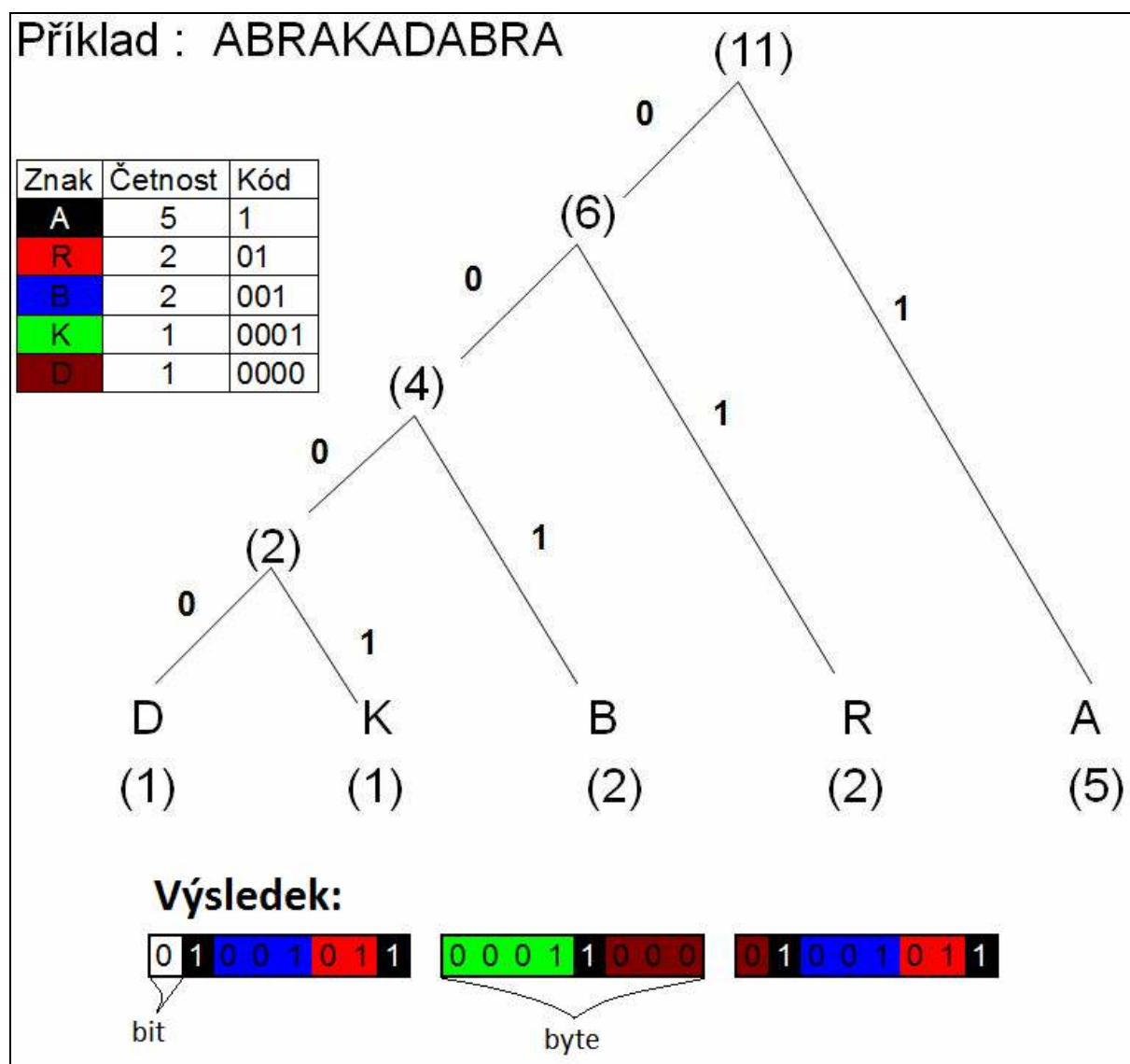


Obrázek 1. Možné podoby Huffmanových stromů [3]

Toto kódování využívá principu binárních stromů. Zde tvořený strom se potom označuje za Huffmanův strom [3]. Od nespécifikovaného binárního stromu se liší svou strukturou, která je uzpůsobena tomu, aby strom přiděloval kódy, jejichž použití poskytne optimální kompresi. Důsledkem toho je skutečnost, že v Huffmanově stromě najdeme znak kódovaný jediným bitem jen v případě, že se ve vstupním řetězci malá skupina znaků vyskytuje s výrazně větší pravděpodobností než zbytek abecedy. Při použití takového postupu totiž s klesající pravděpodobností výskytu znaku velmi rychle roste délka jeho kódu. To je znázorněno na obrázku 1 (b) a pro svou jednoduchost je tento způsob kódování použit i v níže uvedeném příkladu.

Z podoby obou zde znázorněných stromů můžeme usuzovat rozložení pravděpodobností jejich vstupních dat. Vstupní data stromu (a) obsahují znaky, jejichž pravděpodobnost výskytu se vzájemně příliš neliší, každý znak je tak reprezentován stejně dlouhým kódem. Ten by měl být stále kratší, než původní forma zápisu znaku. Strom (b) naopak reprezentuje už zmíněný případ, kdy některé znaky mají značně větší pravděpodobnost výskytu než jiné. Délka kódu pro jednotlivé znaky s jejich klesající pravděpodobností výskytu poměrně rychle roste.

Tento rychlý růst je dán tím, že kódy pro jednotlivé znaky jsou voleny tak, aby kód žádného znaku nebyl součástí kódu jiného znaku, jde o takzvaný prefix kód. To je nezbytnou podmínkou pro umožnění jednoznačné dekomprese, jelikož algoritmus nepracuje s jednotnou délkou slova. Zároveň se algoritmus snaží zvolit takový kód, který zajistí pokud možno nejlepší vlastnosti komprese.



Obrázek 2. Příklad Huffmanova kódování. Převzato z [5] a upraveno.

2.3.1 Příklad Huffmanova kódování

Pro příklad Huffmanova kódování viz obrázek 2. Na příkladu je ukázán postup algoritmu na vstupním řetězci „ABRAKADABRA.“ Jasně je vidět, že nejčastějším symbolem v řetězci je „A“, za ním následují „B“ a „R“, které mají oba shodnou četnost výskytu. Přestože oba tyto symboly by měly být zařazeny stejně, algoritmus jednoduše jeden z nich upřednostní. Stejná situace nastává u posledních dvou symbolů, není to ale žádný problém. Postup přiřazování kódů jednotlivým znakům je znázorněn na binárním stromě. Číslo v závorce nad každým uzlem udává celkovou četnost všech znaků, které uzel řídí. Výsledek je na obrázku znázorněn dole a rozdělen do bytů. Vidíme, že bity kódující znak „D“ přesahují z druhého bytu do třetího, zatímco úplně první bit s bílým pozadím je na obrázku jen do počtu. V tabulce vlevo jsou přehledně vidět kódy všech znaků.

Díky kompresi se velikost vstupu zredukovala z jedenácti na tři byty. Při stanovování kompresního poměru je ovšem nutno počítat s tím, že ke komprimovaným datům je pro úspěšnou dekompresi nutno přiložit tabulku s kódy všech znaků. Výsledný kompresní poměr by proto byl menší.

2.4 Move-to-Front kódování

Název tohoto postupu by se dal přeložit jako „přesun dopředu“, běžně se ale nepoužívá. Anglický název – především jeho zkratka MTF – je už zažitý a český ekvivalent se do hlubšího povědomí zřejmě hned tak nedostane.

MTF kódování se poprvé objevilo v roce 1986. Stojí za ním tým pod vedením Jona L. Bentleyho [6]. Jedná se o lokálně adaptivní kódování, které využívá toho, že pravděpodobnost výskytu jednotlivých znaků ve vstupním řetězci se může měnit. Znakem zde stejně jako u Huffmanova kódování může být jak skutečný symbol, tak celé slovo. S Huffmanovým kódováním má tento postup společný i dvojí průchod vstupními daty.

Princip MTF je jednoduchý. Při prvním průchodu souborem dat je vytvořena abeceda. Seznam všech symbolů, které ve vstupu vyskytují. Tento seznam je abecedně seřazen a v této podobě je uložen, bude totiž dále zužitkován.

Při druhém průchodu je každý znak nahrazen indexem označujícím pozici, na níž se v abecedě se nachází. Před přesunem k dalšímu znaku je navíc aktuální znak v abecedě umístěn na začátek celé abecedy. Frekventovanější znaky budou tedy na začátku a v souboru proto budou nahrazovány indexy o nižší hodnotě. Méně časté znaky budou s největší pravděpodobností na konci abecedy. [3]

Komprimovaná data potom obsahují vektor indexů a abecedu v jejím původním pořadí. Proces dekomprese probíhá v zásadě stejně. Indexy jsou zpětně nahrazeny znaky, během čehož jsou právě použité znaky opět přefazeny na začátek abecedy. Dekodér tak provádí prakticky stejnou operaci jako kodér, tentokrát ale stačí už jen jeden průchod daty. Abeceda je totiž poskytnuta kodérem.

Pokud by se vynechal krok přefazování právě zpracovávaného znaku na počátek abecedy, vektor indexů by měl stejný rozsah. Při vhodné podobě vstupního řetězce je ale průměrná hodnota indexů se zařazeným přefazováním menší [3]. V případě nevhodné podoby vstupu může být situace opačná. Pro efektivní použití MTF musí být proto splněna podmínka výskytu lokálně se opakujících znaků ve vstupní řetězci [3].

Zápis znaků pomocí jejich indexů, tedy jednoduchých čísel, může sám o sobě zajistit jistý stupeň komprese. Kromě toho se ale využívá skutečnosti, že vektor indexů na výstupu kodéru obsahuje větší koncentraci malých čísel, než čísel velkých. Z toho těží Huffmanovo nebo aritmetické kódování, která se za MTF často zařazují.

2.4.1 Příklad MTF kódování

Jako vstupní řetězec pro ilustraci MTF kódování poslouží slovo LOSOS. Znakem budu uvažovat skutečně jeden znak. Postup komprese i dekomprese je znázorněn na obrázku 3.

Komprese			Dekomprese		
<i>Vstup</i>	<i>Abeceda</i>		<i>Vstup</i>	<i>Abeceda</i>	
LOSOS	LOS		12322	LOS	
Krok	Index	Abeceda	Krok	Index	Abeceda
1	1	LOS	1	1	LOS
2	2	OLS	2	2	OLS
3	3	SOL	3	3	SOL
4	2	OSL	4	2	OSL
5	2	SOL	5	2	SOL
<i>Výstup: 12322</i>			<i>Výstup: LOSOS</i>		

Obrázek 3. Příklad MTF kódování

Na obrázku 3 *Krok* udává pozici právě zpracovávaného znaku ve vstupním řetězci, *Index* je jeho pořadí v aktuální podobě abecedy a *Abeceda* je pořadí znaků v abecedě po přesunutí právě zpracovávaného znaku na její začátek.

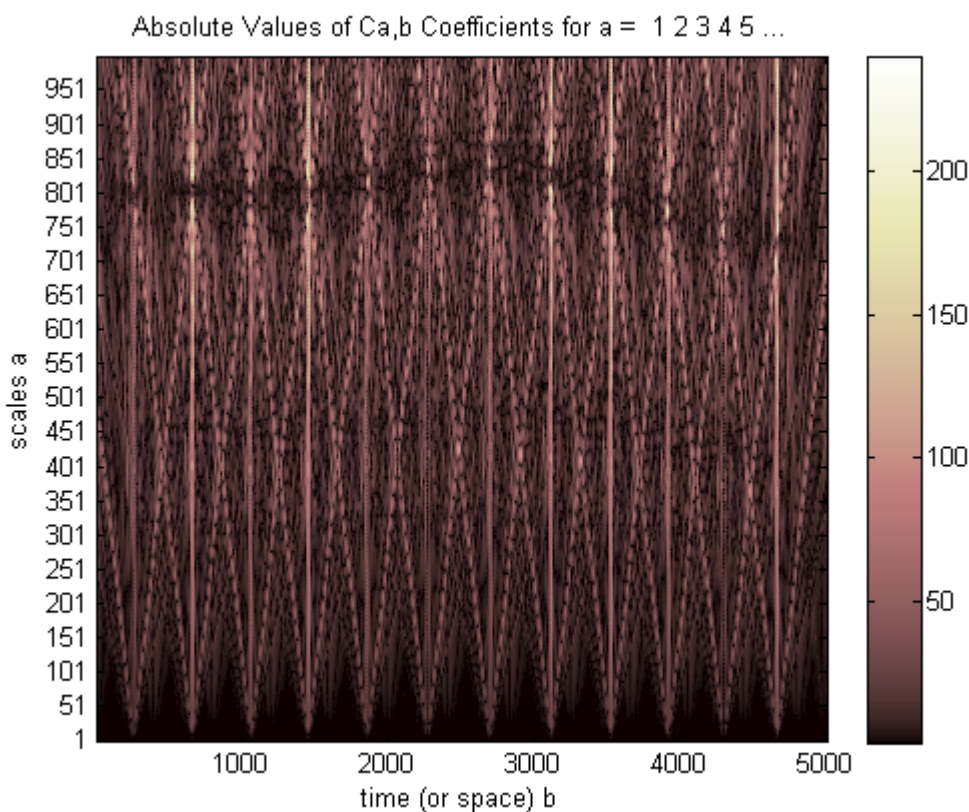
Zatímco při kompresi hledá algoritmus právě zpracovávaný znak v abecedě a jeho index odesílá na výstup, během dekomprese odesílá na výstup znak, který se nachází

na pozici udané právě zpracovávaným znakem na vstupu dekomprese (indexem). Můžeme si všimnout, že pořadí znaků v abecedě je v jednotlivých krocích stejné jak pro kodér, tak pro dekodér. Oba algoritmy pracují ve shodě.

2.5 Vlnková transformace

Vlnková transformace je prostředek pro transformaci signálu z časově-amplitudové prezentace do časově-frekvenční. Umožňuje proto na rozdíl od hojně používané Fourierovy transformace (frekvenčně-amplitudová prezentace) zachytit nejen frekvence obsažené v signálu, ale také konkrétní čas, kdy se v signálu vyskytly.

Přestože první vlnka je přisuzována maďarskému matematikovi Alfrédu Haarovi (1885-1933), pojem vlnková transformace se objevil až v osmdesátých letech dvacátého století, kdy jej zavedl Jean Morlet. Jedná se proto o relativně novou metodu zpracování signálů.



Obrázek 4. Vlnková reprezentace EKG signálu

2.5.1 Spojitá vlnková transformace

Transformaci je podle [7] matematicky možno popsat následujícím vzorcem:

$$CWT_x^\psi(\tau, s) = \frac{1}{\sqrt{|s|}} \int x(t) \psi^* \left(\frac{t - \tau}{s} \right) dt, \quad (1)$$

kde ψ je transformační funkce označovaná jako mateřská vlnka, $x(t)$ je původní signál, τ je translace (posunutí vlnky v čase) a s je měřítko. Vstupní signál je postupně pronásobován s mateřskou vlnkou, která je dilatována a posouvána v závislosti na parametrech měřítka a translace.

Výstup spojitě vlnkové transformace je potom zobrazován ve 3D grafu, kde jednou osou je měřítko, jenž je nepřímo úměrné frekvenci, druhou osou je translace odpovídající časové ose a třetí osou je amplituda. Pro zjednodušení se používají i 2D grafy, kde amplituda je dána barevnou škálou. Toto zobrazení je ilustrováno na obrázku 4.

Konkrétní aplikace vlnkové transformace v procesu komprese bude rozebrána v kapitole 5.

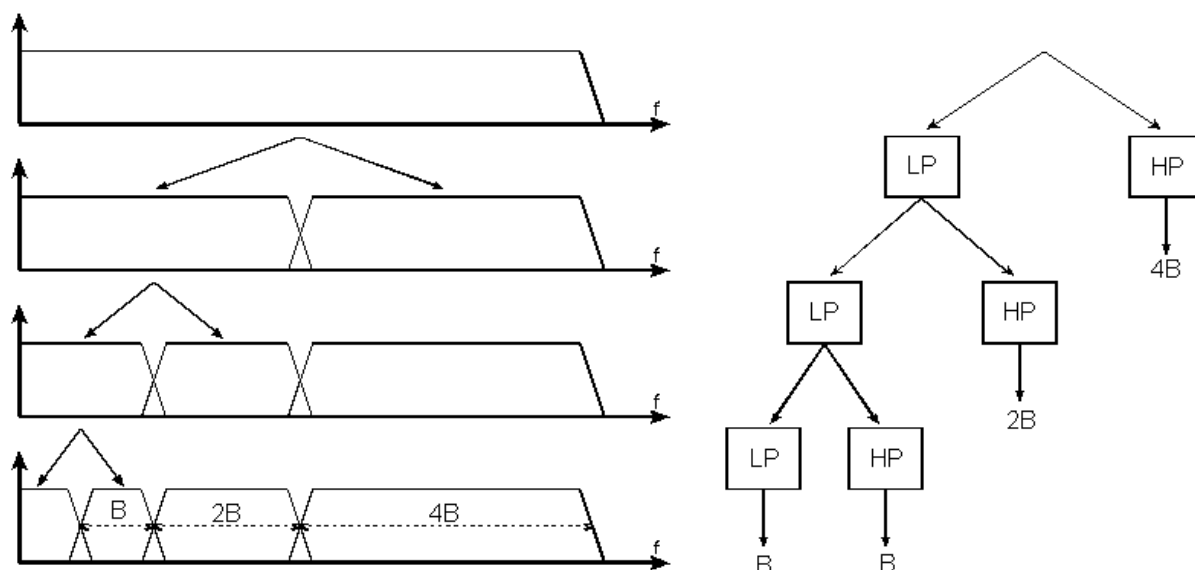
2.5.2 Diskrétní vlnková transformace

Realizace spojitě vlnkové transformace s využitím výpočetní techniky má relativně velké nároky na výpočetní čas. Z hlediska rekonstrukce původního signálu navíc obsahuje zbytečně redundantní informace [8]. Proto se mnohem častěji využívá diskrétní vlnková transformace.

Diskrétní vlnková transformace je realizována frekvenčním rozkladem pomocí digitálních filtrů. V každém kroku jsou na signál aplikovány dva půlpásmové filtry. Horní a dolní propust s mezním kmitočtem ve středu frekvenčního pásma vstupního signálu. Tyto rozkladové filtry musí splňovat určité podmínky po jejichž splnění bývají označovány jako kvadraturní zrcadlové filtry. Jejich modulové charakteristiky jsou symetrické a protínají se právě na mezním kmitočtu. [8]

Dolní propustí projdou pouze frekvence nižší než mezní frekvence. Signál po průchodu filtrem tak má stejný počet vzorků ale nese informace jen o polovičním frekvenčním pásmu. Proto je možné výsledný signál bez ztráty informace podvzorkovat, aby obsahoval jen poloviční počet vzorků. Stejný proces podvzorkování probíhá i u pásma vyšších frekvencí.

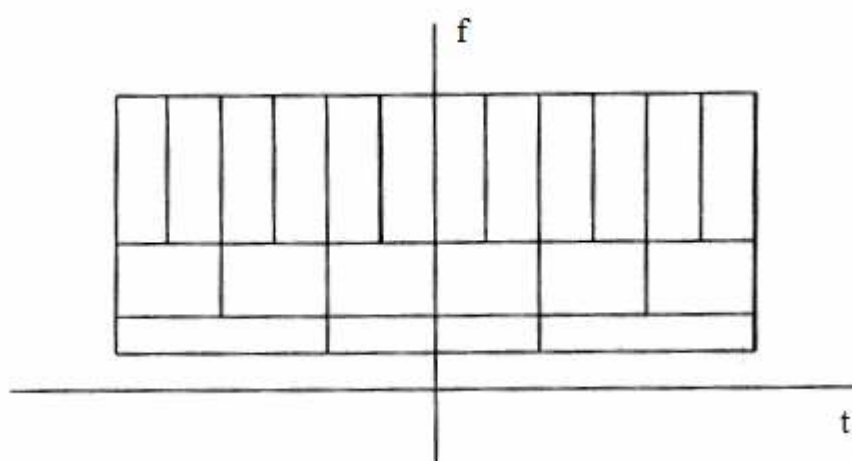
Signál je filtrován tak dlouho, dokud není dosaženo předem daného počtu pásem. Přitom platí, že počet pásem je o jedno více, než kolikrát byly aplikovány filtry.



Obrázek 5. Znázornění frekvenčního rozkladu a uspořádání rozkladových filtrů [9]

Celý postup je ilustrován na obrázku 5. *LP* zde představuje filtr dolní propusti, *HP* filtr horní propusti, B je šířka nejužšího frekvenčního pásma. Pásmo nižších frekvencí je zde dále filtrováno další dvojicí filtrů s mezní frekvencí ve středu tohoto pásma. Po každé filtraci je opět prováděno podvzorkování (blok této operace není na obrázku samostatně znázorněn). Nejvyšší pásmo má poloviční počet vzorků vstupu a každé nižší se dále půlí. To je znázorněno v levé polovině obrázku 5.

Na obrázku je sice děleno pásmo nízkých frekvencí. Obecně to však může být naopak, nebo mohou být dělena pásma obě.

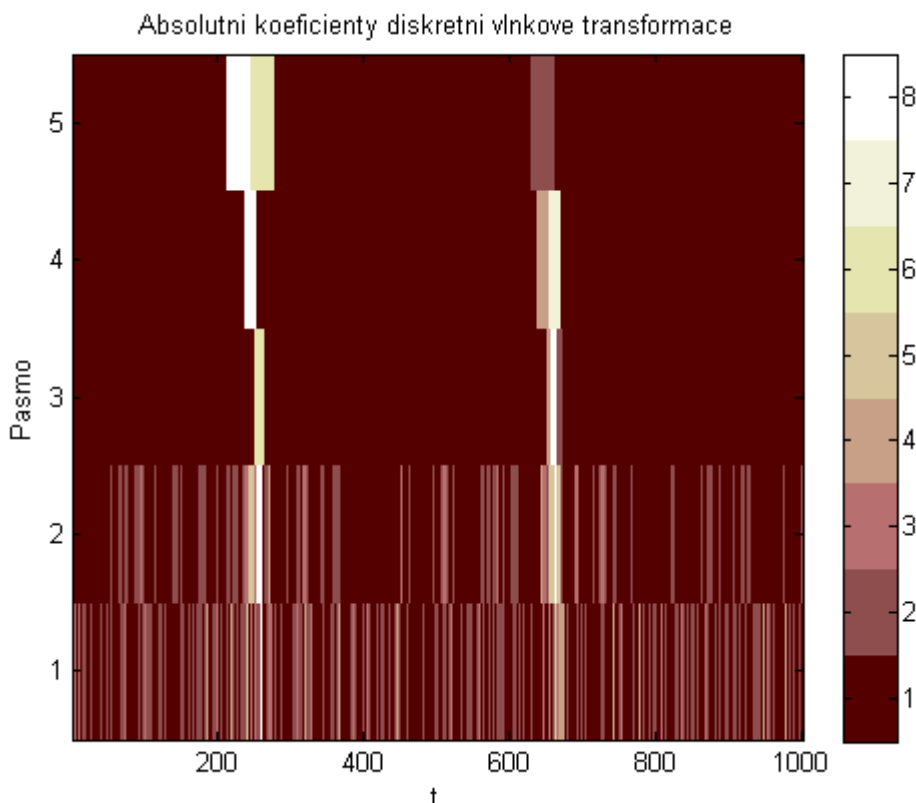


Obrázek 6. Znázornění rozlišení v jednotlivých pásmech [8]

Množství vzorků odpovídá časovému rozlišení daného pásma. Vysoká pásma obsahující největší počet vzorků proto mají nejlepší časové rozlišení. Nízká pásma mají sice malý počet vzorků, ty ale reprezentují mnohem menší rozpětí frekvencí. Mají proto lepší frekvenční rozlišení. Toto rozložení ilustruje obrázek 6. Všechny složky signálu spadající

svými parametry do jednoho pole obrázku nejsou po transformaci vzájemně rozeznatelné. Delší strana pole proto znamená menší rozlišení ve veličině reprezentované příslušnou osou. Plocha všech polí je přitom stejná a odvozuje se od Heisenbergova principu neurčitosti.

Reálný výstup diskretní vlnkové transformace je ukázán na obrázku 7. Jde o část stejného signálu jako na obrázku 4. Můžeme si zde všimnout, že časové rozlišení pásma s nejnižšími frekvencemi (na obrázku pásma 5) je skutečně mnohem horší, než u pásma s nevyššími frekvencemi (na obrázku pásma 1).



Obrázek 7. Příklad vizualizace koeficientů diskretní vlnkové transformace

2.6 Burrows-Wheelerova transformace

Burrows-Wheelerova transformace (BWT) byla prezentována v roce 1994 pány Michaellem Burrowsem a Davidem J. Wheelerem [10]. Využití tohoto postupu bylo od prvopočátku cíleno na kompresi textu. V popisu jejího principu budu proto zmiňovat jako vstupní řetězec právě text. Nic ovšem nevylučuje použít transformaci pro kompresi sekvence čísel.

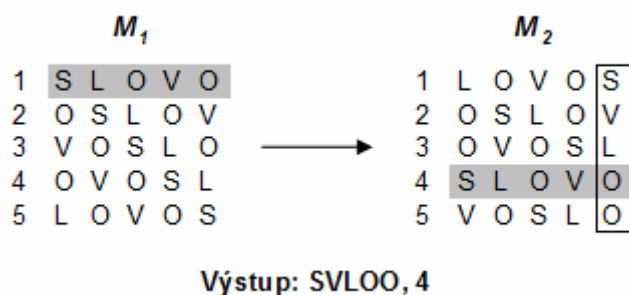
Během postupu transformace je vytvořena matice permutací znaků vstupního řetězce tak, že každý řádek matice představuje jednu permutaci. Permutace jsou realizovány jednoduchým cyklováním. Poslední znak předchozí permutace je umístěn před první.

Výsledkem je potom čtvercová matice obsahující všechny permutace, které takto mohou být získány. Řádky matice jsou poté abecedně seřazeny. Výstupem celé transformace je poslední sloupec takto vytvořené matice a index řádku, na kterém se nachází původní řetězec v nezměněném pořadí.

Výstup je proto o jeden znak delší, než vstup. Pořadí znaků je v něm ale změněno tak, aby aplikace dalších kompresních postupů docílila lepšího kompresního poměru. V ideálním případě seskupuje BWT stejné znaky k sobě. To je dáno pravděpodobností, s jakou po sobě tyto znaky v jednotlivých jazycích následují.

2.6.1 Příklad Burrows-Wheelerovy transformace

Obrázek 8 ilustruje právě popsany postup. Vstupem je řetězec SLOVO, na obrázku podbarven šedě. Matice M_1 obsahuje permutace v takovém pořadí, v jakém byly tvořeny. V matici M_2 jsou už tyto permutace seřazeny abecedně. Vstupní řetězec se zde nachází na čtvrtém řádku, proto je výstupem kromě posledního sloupce M_2 právě index čtyři. Na příkladu je jasně vidět seskupování stejných znaků dohromady, oba znaky „O“ jsou nyní vedle sebe. Přestože právě toto chování je důvodem použití Burrows-Wheelerovy transformace pro účely komprese, tento jev nenastává vždy a rozhodně nelze tvrdit, že by transformace seskupila všechny stejné znaky vstupu za sebe.

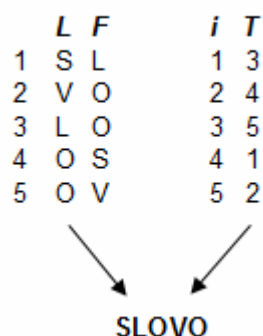


Obrázek 8. Příklad Burrows-Wheelerovy transformace

2.6.2 Příklad inverzní Burrows-Wheelerovy transformace

Pro inverzní transformaci je k dispozici pouze poslední sloupec původní matice a index řádku, na němž se nachází originální řetězec. V našem případě tedy řetězec SVLOO a index 4. Označme tento řetězec jako L . Seřazením jeho znaků podle abecedy, získáme vektor F . Ten se shoduje s prvním sloupcem matice M_1 . V průběhu řazení je důležité sledovat, z jaké pozice vektoru L jednotlivé znaky pocházejí. Takto získaný vektor pozic (na obrázku 9

označený jako T) je klíčem k uskutečnění inverzní Burrows-Wheelerovy transformace. Indexy prvků tohoto vektoru jsou označeny jako i .



Obrázek 9. Příklad inverzní Burrows-Wheelerovy transformace

Nyní přichází na řadu index řádku obsahujícího v matici M_2 původní řetězec. To byl v našem případě čtvrtý řádek. Když se podíváme na čtvrtou pozici vektoru T , je zde číslo 1. To nám říká, že první znak původního řetězce se nachází na první pozici vektoru L . Abychom našli další znak, musíme se opět vrátit k vektoru T a zjistit, jaký prvek je právě na první pozici. Je to prvek 3, druhý znak původního řetězce je proto na třetí pozici vektoru T . Další postup je analogický. Matematicky jej lze popsat následující rovnicí [3]:

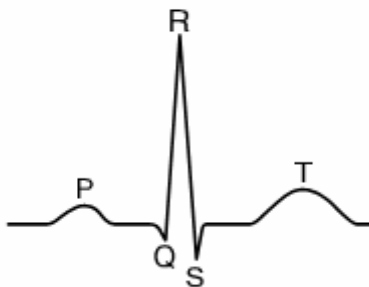
$$L(T(i)) \leftarrow L(i) \quad (2)$$

Šipka v rovnici (2) naznačuje, že $L(T(i))$ vždy stojí před $L(i)$. To je základní úvaha vedoucí k postupu použitému v předchozím odstavci. Všechny symboly zde mají stejný význam jako na obrázku 9.

Pro efektivní činnost potřebuje transformace relativně dlouhý vstupní řetězec. Uvažuje se řádově alespoň několik kilobytů [10]. Teprve potom je možno očekávat, že ve výstupu se budou nacházet delší skupiny opakujících se znaků. Ty je následně možné komprimovat například proudovým kódováním a očekávat mnohem lepší kompresní poměr než by mohl být dosažen u vstupních dat v jejich původní podobě.

3 EKG

Řízení srdečních stahů funguje na principu přenosu budícího napětí drahami v srdeční svalovině. Není to ovšem toto budící napětí, které snímáme jako EKG (elektrokardiogram). Snímané napětí je výsledkem superpozice akčního napětí generovaného každým kardiomyocytem při vykonávání mechanické práce – stahu. Toto napětí je mnohem vyšší, než budící napětí. Při snímání pomocí končetinových svodů je napětí od vrcholu R vlny po vrchol S vlny až 1,5 mV. Pro průběh EKG kmitu viz Obrázek 10. Velikost napětí je ale vysoce závislá na umístění elektrod. Pokud by jedna elektroda byla umístěna přímo nad srdcem a druhá jinde na těle, daleko od srdce, snímané napětí by mohlo dosáhnout až čtyř milivoltů [11].



Obrázek 10. Průběh jedné periody idealizovaného EKG. Převzato z [12].

Signál EKG je kvaziperiodický. Znamená to, že se po sobě s určitou periodou opakuje na první pohled podobné QRS komplexy. Tyto komplexy se od sebe ale liší, a to i u zcela zdravého jedince v absolutním klidu. To je dáno jednak fyziologicky, jednak přítomností rušení a podstatou vzorkování signálu. Ze zřejmých důvodů není ani perioda těchto komplexů v čase konstantní.

3.1 Objem snímaných dat

Standardní EKG signál je snímán třemi bipolárními končetinovými svody. Kromě těchto svodů se však používají i tři zesílené unipolární končetinové svody. Rozdíl je v zapojení elektrod, pro účely komprese a stanovení objemu snímaných dat jsou to jen další tři kanály. Často je EKG snímáno také pomocí šesti hrudních svodů. Kromě zmíněných existují i jiné svodové systémy, například ortogonální svodový systém. Ty ale nejsou tak rozšířené.

Každý svod generuje data, která jsou dále digitalizována. Vzorkovací frekvence takových dat se podle povahy vyšetření pohybuje obvykle od 125 Hz do 500 Hz [13]. Pokud snímáný signál EKG obsahuje impulsy kardiostimulátoru, je nutno použít výrazně vyšší vzorkovací frekvenci. V takových případech se vzorkuje frekvencí až 2000 Hz. Amplituda signálu se kvantizuje s použitím osmi až dvanácti bitů pro zápis každého vzorku [13].

V extrémním případě, kdy snímáme data ze všech dvanácti svodů při vzorkovací frekvenci 2000 Hz a hloubce 12 bitů, dostáváme datový tok 288 kbit/s. To znamená, že každou minutu se nám na pevný disk bez použití jakékoliv formy komprese zapíše 2,16 MB, za hodinu je to potom 129,6 MB. Taková velikost představuje problém zvláště při přenosu těchto dat přes síť.

Záznamy EKG signálů se přitom nemusí omezovat jen na dobu několika málo hodin. V případě ambulantního monitorování EKG, kdy se používají přenosné osobní přístroje nazývané Holtery, je signál snímán po dobu řádově 24 hodin. Přitom i holtery musí být schopny snímat signál s impulsy kardiostimulátoru, proto používají i jinak značně nadsazenou vzorkovací frekvenci 2000 Hz. Lékař navíc může požadovat, aby takové měření probíhalo vícekrát, a mohl tak porovnávat záznamy z jednotlivých dnů [13]. Nutno však podotknout, že při ambulantním monitorování EKG se nevyužívá všech dvanáct svodů. Holterovskými funkcemi jsou dnes vybaveny dokonce i některé kardiostimulátory [14].

4 Implementace proudového kódování pro EKG

Proudové kódování je pravděpodobně nejjednodušším způsobem komprese. Využívá ho většina sofistikovanějších algoritmů. V praxi se EKG vždy snímá více než jedním senzorem. Jeden soubor s uloženým EKG průběhem se potom skládá z několika EKG signálů uložených v řádcích, popřípadě sloupcích.

Veškerá práce byla prováděna v programovém prostředí MATLAB. Jelikož se umístění indikátoru komprese přímo do komprimovaného signálu ukázalo jako problematické, rozhodl jsem se ukládat pozici vzorku nesoucího počet opakování vzorku po něm následujícím do samostatného pomocného vektoru.

Vlastností programového prostředí MATLAB je výrazně větší rychlost výpočtů pro sloupcově uspořádaná data. Algoritmus proto pracuje právě s takovými daty. Řádkově uspořádaný vstup automaticky transponuje.

Přestože signály jednotlivých svodů jsou spolu značně korelovány [13], stále se jedná o samostatné řetězce dat, které jako samostatné musí vystupovat i po rekonstrukci. Pokud by ale byla při kompresi ponechána stejná struktura dat (matice o počtu sloupců shodném s počtem svodů), došlo by k omezení komprese. Jelikož by data každého svodu musela mít stejný počet vzorků, kompresní poměr celého signálu by byl omezen na kompresní poměr nejhůře zkomprimovaného svodu. Ostatní svody by bylo nutno doplnit na jeho délku. Na obrázku 11 ilustruji tuto situaci červeně podbarvenými nulami.

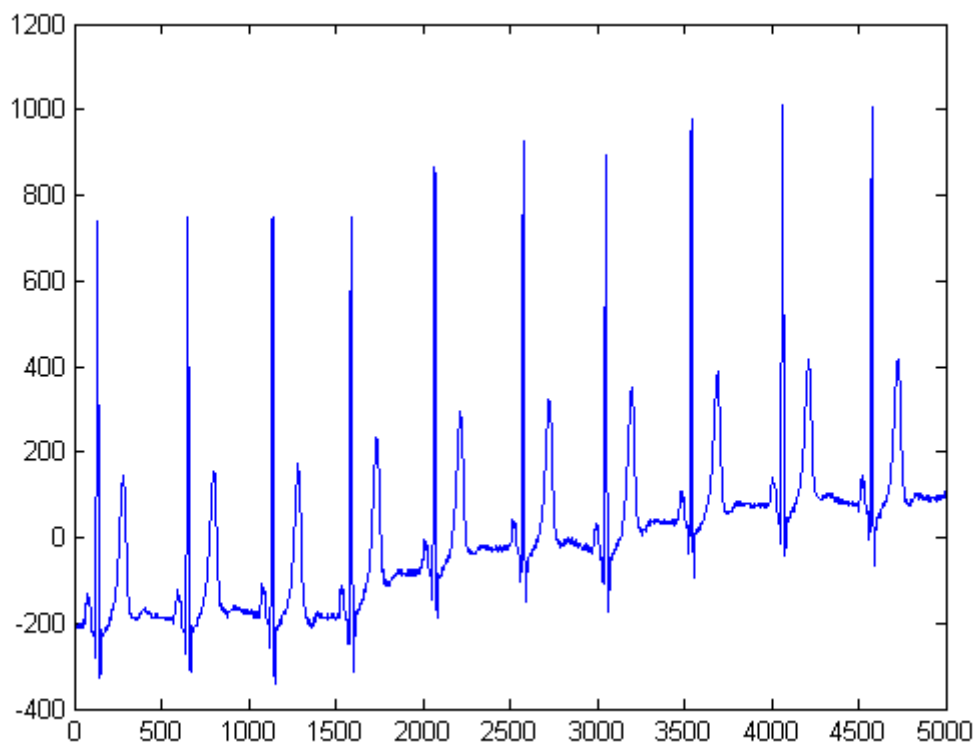
723	308	-141
723	298	-150
718	293	-155
708	293	-150
698	298	-155
703	298	-150
698	308	-145
693	0	-150
693	0	-131
693	0	-121
698	0	0
693	0	0
693	0	0

Obrázek 11. Ukázka doplnění svodů nulami

Na obrázku 11 jsou červeně zvýrazněny nuly nutné k doplnění svodů na jednotnou délku. Tyto nuly ale představují data navíc, ve skutečném EKG záznamu jich kromě toho bývá výrazně více, řádově stovky vzorků s nulovou informační hodnotou. Na obrázku byl

jejich počet omezen tak, aby byla zachována jeho ilustrační hodnota při současném zachování rozumné velikosti. Způsobů řešení tohoto problému je více. V tomto algoritmu jsem se rozhodl zapisovat všechny svody za sebe.

Přesto se jeho přímá aplikace na EKG signál neukázala jako efektivní. To plyne z povahy samotného signálu, příklad reálného signálu je na obrázku 12.



Obrázek 12. Průběh reálného záznamu EKG.

Na obrázku 12 je vidět, že se v reálném signálu EKG nevyskytují delší sekvence opakujících se znaků. Dokonce ani mezi jednotlivými vlnami není signál zcela monotónní. Vždy je do jisté míry přítomno rušení interferující s ideálním průběhem EKG křivky.

4.1 Zhodnocení výsledků

Výše popsaný algoritmus proudového kódování byl přímo aplikován na soubory z CSE Multilead Database (CSE – Common Standards for Quantitative Electrocardiography). Konkrétně na signály datasetu 1 MO1_001_03 až MO1_125_03. Jedná se o třísvodové signály vzorkované při 500 Hz, každý o délce deseti sekund. Každý soubor se tedy skládá z 3×5000 vzorků. Ačkoli databáze obsahuje i uměle vytvořené signály. V této práci používám výhradně reálné průběhy EKG.

Průměrný kompresní poměr dosažený na těchto signálech byl pouze **1,0543**. Použitím tohoto postupu lze tedy dosáhnout pouze mizivé úspory místa. Ta je tak malá, že použití algoritmu v praxi považují za bezpředmětné.

Kompresní poměr byl vypočítán podle vzorce:

$$CR = \frac{SV}{SK}, \quad (3)$$

kde CR je kompresní poměr, SV je velikost vstupních dat a SK je velikost komprimovaných dat (součet velikostí vektoru pozic a vektoru komprimovaných dat). Vzhledem k tomu, že MATLAB při ukládání souborů aplikuje vlastní kompresní postupy, nelze při výpočtu CR použít velikost původních dat souborů. Místo toho je nutno uložit jak data původní, tak data komprimovaná, do MATLABem nijak nekomprimovaných datových souborů a s jejich velikostmi potom pracovat. Do výpočtu CR byly právě proto dosazeny velikosti takto získaných datových souborů.

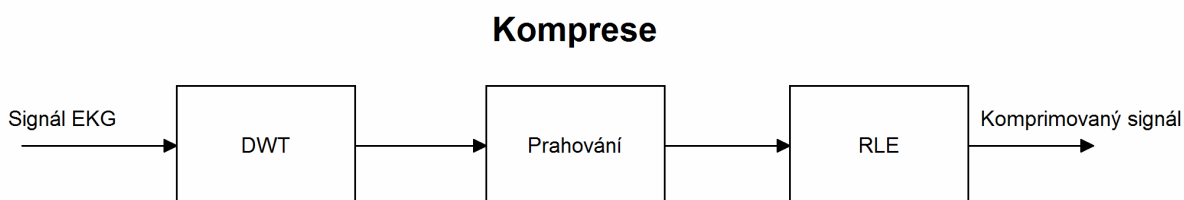
5 Implementace vlnkové transformace

Přímá aplikace proudového kódování nepřinesla znatelnou úsporu místa, zařadil jsem tedy do procesu komprese další operaci. Diskrétní vlnkovou transformaci. Ta je aplikována na každý svod zvlášť. Výstup transformace je opět ukládán do jedné proměnné v podobě matice, vlnkový rozklad všech svodů má totiž stejnou délku. Kromě toho při transformaci vzniká pomocný vektor, ten je ovšem pro všechny svody stejný, stačí proto uložit jen jedinou kopii.

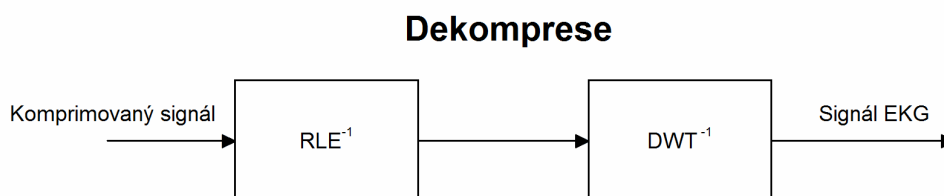
Pouhá aplikace vlnové transformace by kompresní poměr nijak nezlepšila. Právě naopak, zvětšil by se objem vstupních dat. Dalším krokem je proto prahování koeficientů transformace. Prahování je postup, kdy se koeficienty, jejichž hodnota je blízká nule, jednoduše nahradí nulou, matematický zápis je uveden vzorcem:

$$x = \begin{cases} 0 & \text{pro } |x| \leq p \\ x & \text{pro } |x| > p \end{cases}, \quad (4)$$

kde x je hodnota koeficientu vlnkové transformace a p je práh. Vznikají tak dlouhé sekvence nul, což je forma dat velice vhodná pro kompresi s využitím proudového kódování, které je uplatněno jako další krok. Díky aplikaci prahování sice komprese přestává být zcela bezztrátovou, za tu cenu je ovšem dosaženo výrazně lepších kompresních poměrů, viz kapitola 5.3.



Obrázek 13. Diagram procesu komprese signálu



Obrázek 14. Diagram procesu dekomprese signálu

Diagram procesu komprese je zobrazen na obrázku 13, na dalším obrázku se nachází diagram procesu dekomprese. *DWT* značí blok diskrétní vlnkové transformace a *RLE* blok proudového kódování. Horní index -1 označuje inverzní operaci.

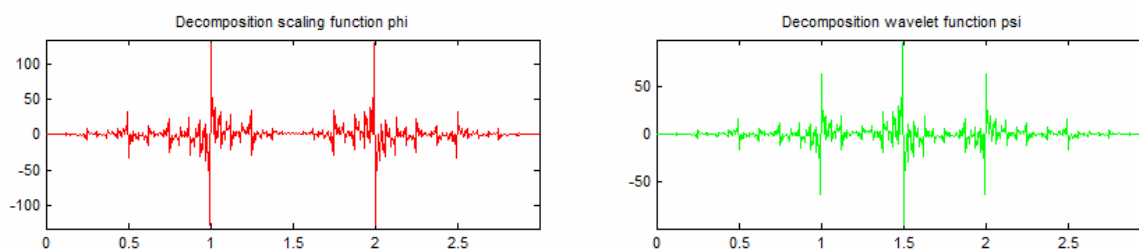
5.1 Modifikace proudového kódování

Na vektor prahovaných koeficientů je uplatněno modifikované proudové kódování. Stejně jako u přímé implementace proudového kódování jsou všechny svody zapisovány za sebe. Jelikož se však mezi koeficienty vyskytuje jen málo opakujících se znaků jiných než nula, nepovažoval jsem za nutné ukládat i pomocný vektor pozic. Předpokladem je, že jediným opakovaným znakem budou nuly a že výskyt ojedinělých nul je velice nízký. Proto za každou nulou následuje vzorek nesoucí počet jejích opakování, i kdyby na dané pozici byla nula jen jedna.

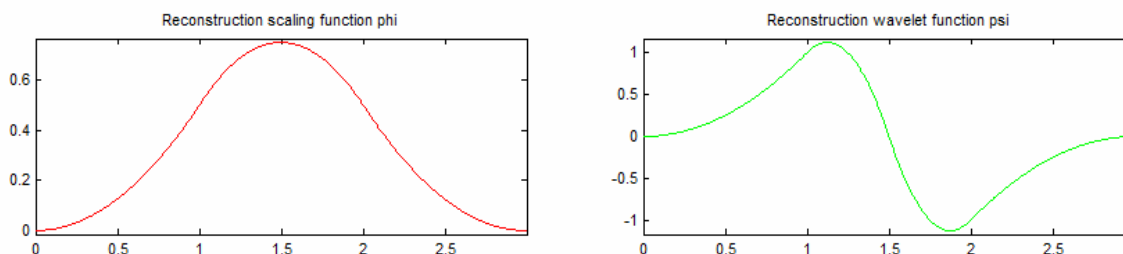
5.2 Parametry vlnkové transformace

Pro co nejlepší kompresi a následnou rekonstrukci signálu je nutné stanovit parametry transformace. Těmi je samotná mateřská vlnka, počet pásem transformace a hodnota prahu.

5.2.1 Vlnka



Obrázek 15. Dekompoziční funkce vlnky bior3.1

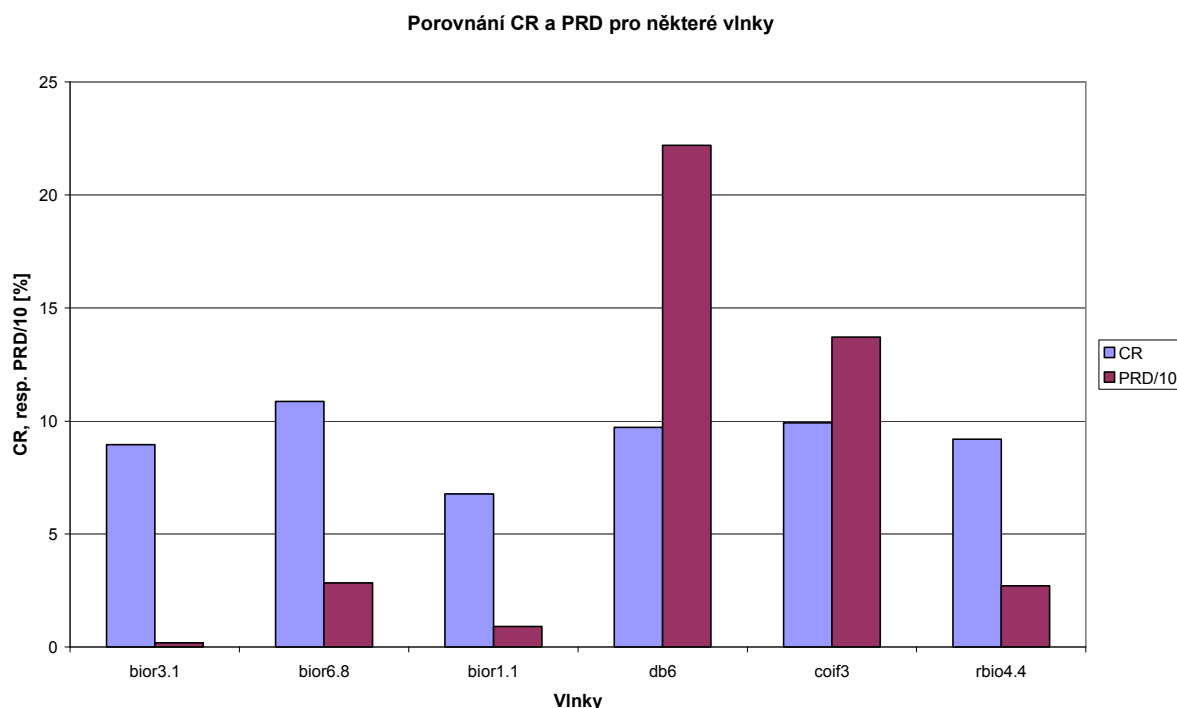


Obrázek 16. Rekonstrukční funkce vlnky bior3.1

Pro výběr vlnky bohužel neexistují žádná exaktní kritéria. Většinou jde o metodu pokus-omyl, nebo intuici jednotlivce. V mém případě jsem nechal proběhnout kompresi s využitím všech vlnek v databázi MATLABu. Z výsledného souboru hodnot se jako nejlepší

ukázala biortogonální vlnka 3.1 (bior3.1). Na obrázcích 15 a 16 jsou zobrazeny dekompoziční a rekonstrukční funkce mateřské vlnky bior3.1. Na těchto obrázcích osa x představuje čas, zatímco osa y měřítko.

Obrázek 17 ukazuje hodnoty CR a PRD pro některé jiné vlnky. Aby byly obě tyto veličiny souměřitelné, bylo nutno zobrazit hodnoty PRD desetkrát menší, než ve skutečnosti jsou. Můžeme si všimnout, že jsou zde zastoupeny i vlnky, s jejichž použitím lze dosáhnout lepšího kompresního poměru (například bior6.8, coif3), to je ale vždy vykoupeno neúměrně větší hodnotou PRD. Tyto údaje jsou platné pro práh $0,3 \sigma$ (σ – směrodatná odchylka signálu) a pět pásem transformace, jde o průměr výsledků všech sto dvaceti pěti zde používaných signálů. Vzhledem k množství vlnek nebylo možné uvést hodnoty pro každou z nich.



Obrázek 17. Hodnoty CR a PRD pro různé vlnky

5.2.2 Počet pásem transformace

Jak již bylo řečeno v kapitole 2.5.2, výpočet vlnkové transformace dělí signál podle frekvencí vždy na dvě části. Tento krok bývá přitom několikrát opakován za vzniku většího počtu pásem. Pásma nižších frekvencí bývají označována jako aproximace, zatímco pásma vyšších frekvencí jako detaily. V této práci budou rozkládána pásma nižších frekvencí, právě ta totiž nesou klíčové informace EKG signálu. V MATLABu se koeficienty všech pásem ukládají do jednoho vektoru. Jejich hranice jsou potom hodnotami pomocného vektoru.

Použití vlnkové transformace samo o sobě přináší zaokrouhlovací chyby. Ty jsou ale v porovnání se zkreslením způsobeným prahováním koeficientů zanedbatelné. Celkové zkreslení se při vhodné volbě parametrů i tak ukazuje jako minimální.

5.2.3 Práh

Na rozdíl od předchozích dvou parametrů, velikost prahu vlastnosti transformace přímo nijak neovlivní. Prahování je totiž aplikováno až na koeficienty transformace po jejím skončení. Velikost prahu udává jak velkou ještě může mít koeficient absolutní hodnotu, aby byl nahrazen nulou. Vyšší práh logicky znamená vyšší kompresní poměr, ale také větší zkreslení signálu.

Nejpřímější provedení prahování by bylo zvolení vhodné konstanty, která by práh reprezentovala. Takový postup ale nezohledňuje různou povahu a hlavně různou dynamiku signálů. Efektivnější je přizpůsobit velikost prahu každému signálu na míru. Proto se zde jeho velikost odvozuje od směrodatné odchylky signálu. Odchylka je počítána pro každý svod signálu zvlášť, a to podle následujícího vzorce:

$$\sigma = \sqrt{\sum_{i=1}^N \frac{(x_i - \bar{x})^2}{N-1}}, \quad (5)$$

kde x_i je i -tý vzorek signálu, \bar{x} je aritmetický průměr hodnot všech vzorků signálu a N je počet vzorků.

5.3 Zhodnocení výsledků

Aplikace vlnkové transformace a následné prahování umožňuje dosažení mnohem vyšších kompresních poměrů, než přímá aplikace proudového kódování. Na druhou stranu se už nejedná o zcela bezztrátovou kompresi. Jak bylo uvedeno výše, vlastnosti komprese jsou závislé na dvou hlavních parametrech. Jde o počet pásem transformace a velikost prahu. Nastavení těchto dvou parametrů závisí na požadavcích, které na kompresi klademe. Jejich přesnější vliv je zhodnocen na následujících stránkách. Pokud není řečeno jinak, všechny hodnoty byly získány jako průměrná hodnota daného parametru u stejných souborů, jako při přímé aplikaci proudového kódování.

5.3.1 Stanovení zkreslení signálu

Odlišnost signálu po dekompresi od jeho původní podoby byla určena pomocí parametru PRD (Percent Root-mean-square Difference). Tato metoda je objektivním hlediskem změny průběhu EKG, protože nerozlišuje důležité úseky signálu od méně

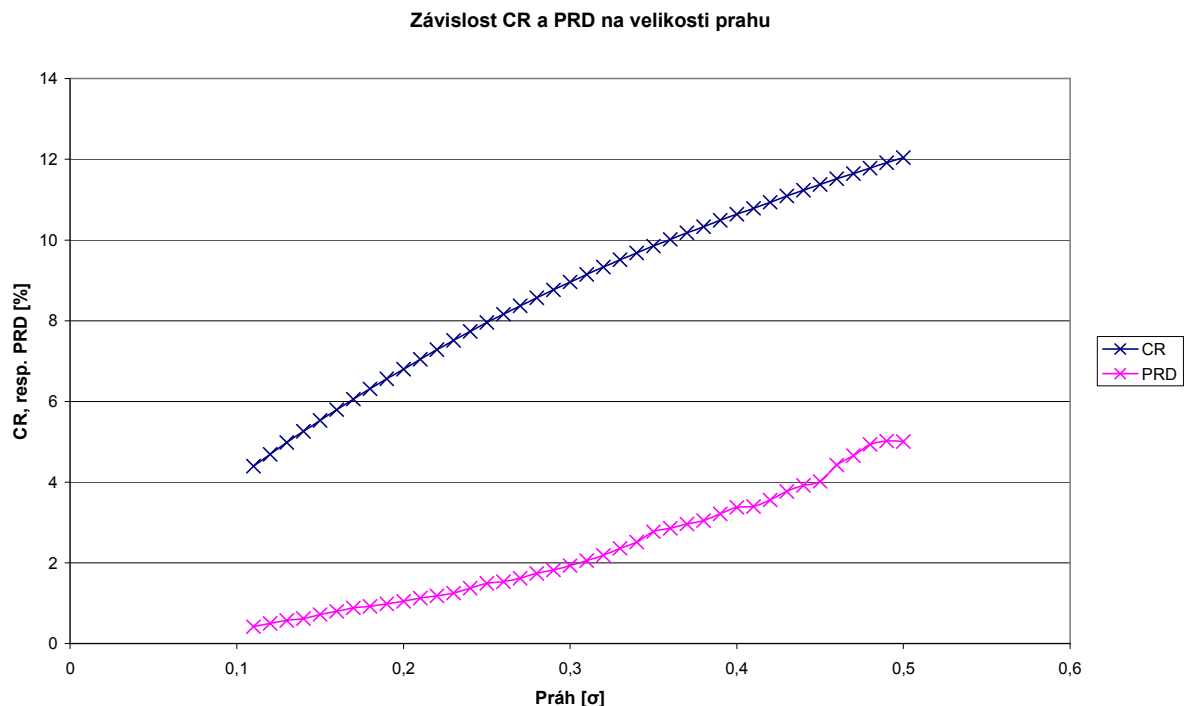
důležitých. Jelikož proces hodnocení EKG je dodnes záležitostí především subjektivní, není absolutní objektivita PRD ideální. Pro diagnózu totiž nemají všechny úseky EKG stejnou důležitost. Přesto zde tento parametr uvádím, protože jeho stanovení je jednoduché a v problematice ztrátové komprese je využíván velmi často [13] [15].

Hodnota PRD je počítána podle vzorce:

$$PRD = \sqrt{\frac{\sum_i (x_i - y_i)^2}{\sum_i y_i^2}} \cdot 100, \quad (6)$$

kde x_i je i-tý vzorek původního signálu a y_i je i-tý vzorek signálu po rekonstrukci.

Cílem je samozřejmě dosáhnout minimálního zkreslení při maximálním kompresním poměru. Jako největší přípustné zkreslení uvažuji hodnotu PRD 2 %. Kromě toho zde budou uvedeny i CR pro hodnotu PRD 1 %.



Obrázek 18. Závislost CR a PRD na hodnotě prahu

5.3.2 Vliv velikosti prahu

Na obrázku 18 vidíme, že se zvyšujícím se prahem podle očekávání roste jak kompresní poměr, tak hodnota PRD. Osa x v grafu představuje násobek směrodatné odchylky. Velikost prahu je potom určena vzorcem:

$$p = k \cdot \sigma, \quad (7)$$

kde p je hodnota prahu, σ je směrodatná odchylka daného svodu signálu, k zastupuje díl odchylky započítaný jako práh. Hodnoty, z nichž graf vychází, jsou obsaženy v tabulce 1.

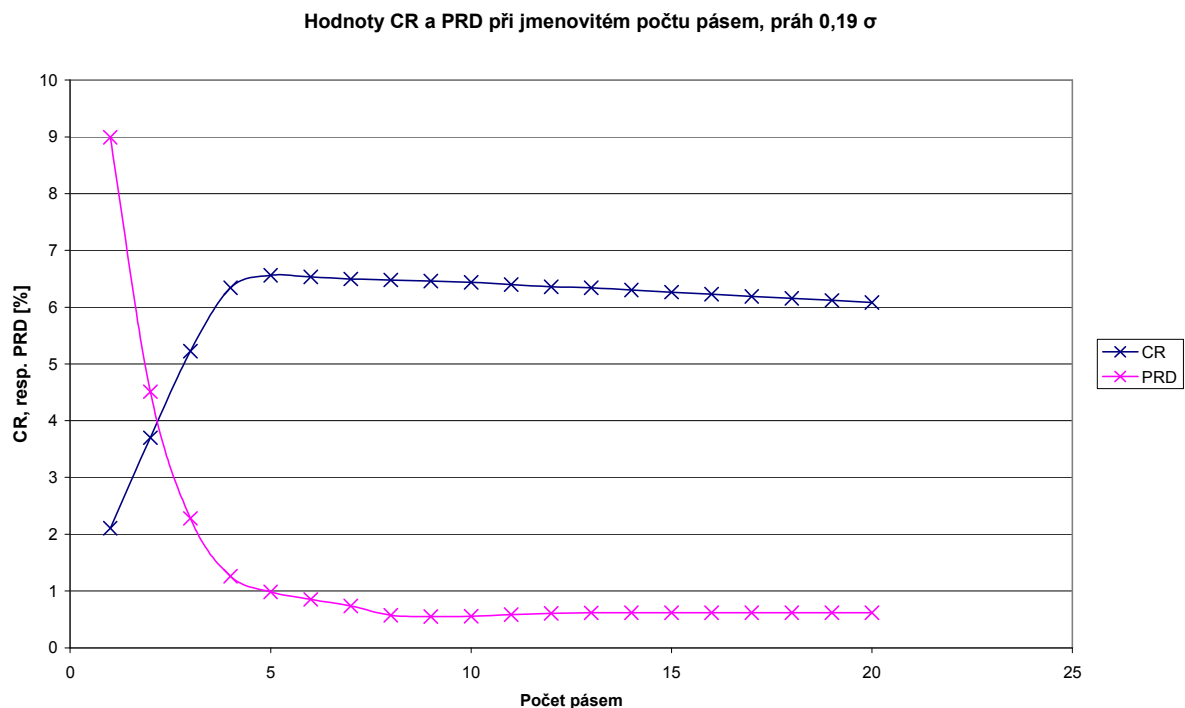
Hodnota PRD při prahu rovnu 0,3 násobku směrodatné odchylky je poslední hodnotou pod hranicí dvou procent. Pod jedním procentem se drží při prahu 0,19 σ a menším. Tato data byla získána při rozkladu signálu na pět frekvenčních pásem, to je počet, při kterém je dosaženo nejlepší komprese, jak je uvedeno v následující kapitole. Kompresní poměr je zde stejně jako ve všech jiných grafech počítán ze vztahu (3).

Tabulka 1. Hodnoty CR a PRD pro jmenovité hodnoty prahu

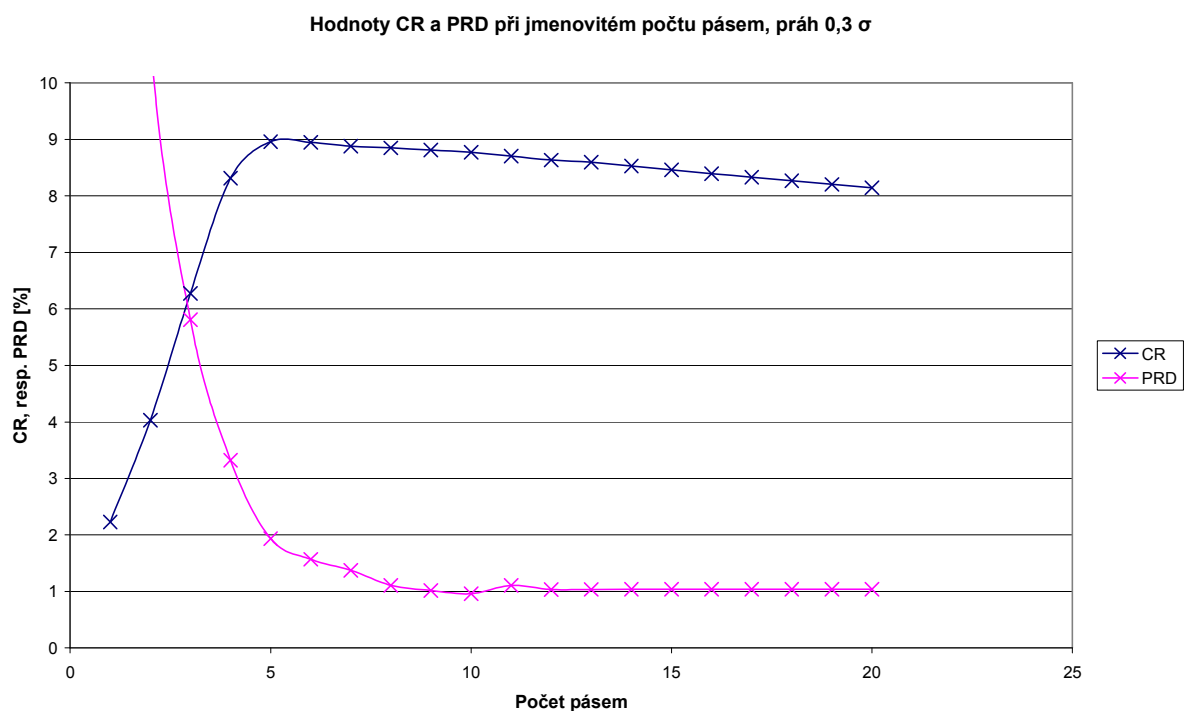
Práh	CR	PRD [%]	Práh	CR	PRD [%]
0,11	4,3965	0,4492	0,36	10,0161	2,8701
0,12	4,6904	0,5231	0,37	10,1774	2,9762
0,13	4,9833	0,5913	0,38	10,3316	3,0678
0,14	5,2634	0,6374	0,39	10,4869	3,2361
0,15	5,5289	0,7346	0,40	10,6425	3,3974
0,16	5,7957	0,8116	0,41	10,7855	3,4169
0,17	6,0589	0,8896	0,42	10,9358	3,5839
0,18	6,3132	0,9348	0,43	11,0894	3,7936
0,19	6,5601	0,9920	0,44	11,2360	3,9385
0,20	6,8013	1,0657	0,45	11,3777	4,0345
0,21	7,0454	1,1410	0,46	11,5182	4,4451
0,22	7,2804	1,1866	0,47	11,6465	4,6834
0,23	7,5132	1,2554	0,48	11,7840	4,9598
0,24	7,7349	1,3775	0,49	11,9140	5,0420
0,25	7,9588	1,4979	0,50	12,0466	5,0385
0,26	8,1618	1,5459	0,51	12,1654	5,2331
0,27	8,3703	1,6257	0,52	12,2935	5,4438
0,28	8,5677	1,7489	0,53	12,4172	5,5576
0,29	8,7634	1,8318	0,54	12,5329	5,5828
0,30	8,9612	1,9355	0,55	12,6496	5,7690
0,31	9,1470	2,0537	0,56	12,7562	5,9255
0,32	9,3319	2,1858	0,57	12,8609	6,1064
0,33	9,5108	2,3616	0,58	12,9719	6,1705
0,34	9,6792	2,5244	0,59	13,0756	6,4116
0,35	9,8514	2,7839	0,60	13,1805	6,4862

5.3.3 Vliv počtu pásem

Na obrázku 19 vidíme, že pro práh 0,19 σ se po dosažení čtyř pásem již kompresní poměr příliš nemění, přičemž jeho nejvyšší hodnoty dosahujeme pro pět pásem, a to **6,5601**. Při dalším zvyšování počtu pásem se kompresní poměr velmi zvolna zmenšuje. Zároveň se ale zmenšuje i hodnota PRD, a to o poznání rychleji. Pro pět a více pásem je průměrná hodnota PRD pod jedním procentem. Pokles PRD se ale z přibývajícími pásmy zpomaluje a od deseti pásem se tento trend dokonce zvrátí a nastává pomalý nárůst.



Obrázek 19. Závislost CR a PRD na počtu pásem transformace při prahu $0,19 \sigma$



Obrázek 20. Závislost CR a PRD na počtu pásem transformace při prahu $0,3 \sigma$

Obrázek 20 ukazuje, že při prahu $0,3 \sigma$ dosahujeme obecně vyššího kompresního poměru, ale také vyšších hodnot PRD. Hodnoty PRD u nejnižších počtů pásem jsou v tomto případě skutečně markantní. První dvě nejsou ani v rozsahu zobrazovaného grafu,

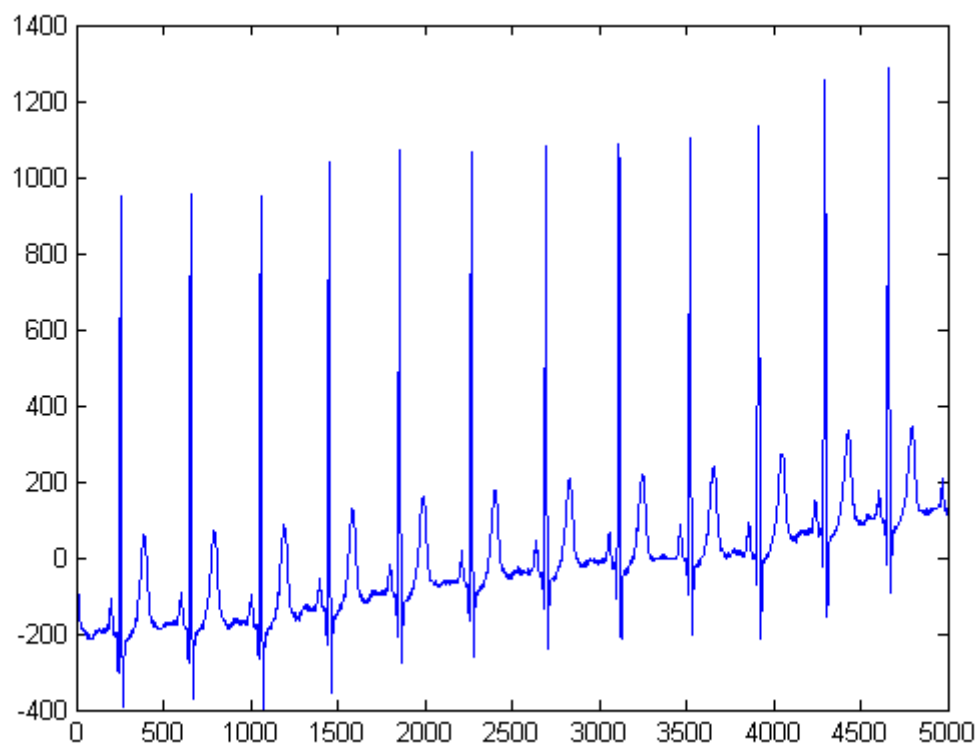
jejich hodnoty je ale možno odečíst z tabulky 2. Kompresní poměr se tentokrát ustálil až při pěti a více pásmech a nejlepší hodnota se opět nachází právě při pěti pásmech, jmenovitě **8,9612**. Pro tuto hodnotu už ale PRD představuje téměř dvě procenta. Kompresní poměr a stejně tak i PRD s dalšími pásmy klesají. Pod jedno procento se ale trvale nedostane. Můžeme pozorovat ustálení hodnoty PRD těsně nad jedním procentem s dalšími pásmy by tato hodnota stejně jako v předchozím případě začala pomalu růst. Tento jev ale nastává mimo zobrazenou oblast.

Tabulka 2. Hodnoty CR a PRD pro různé počty pásem při jmenovitých hodnotách prahu

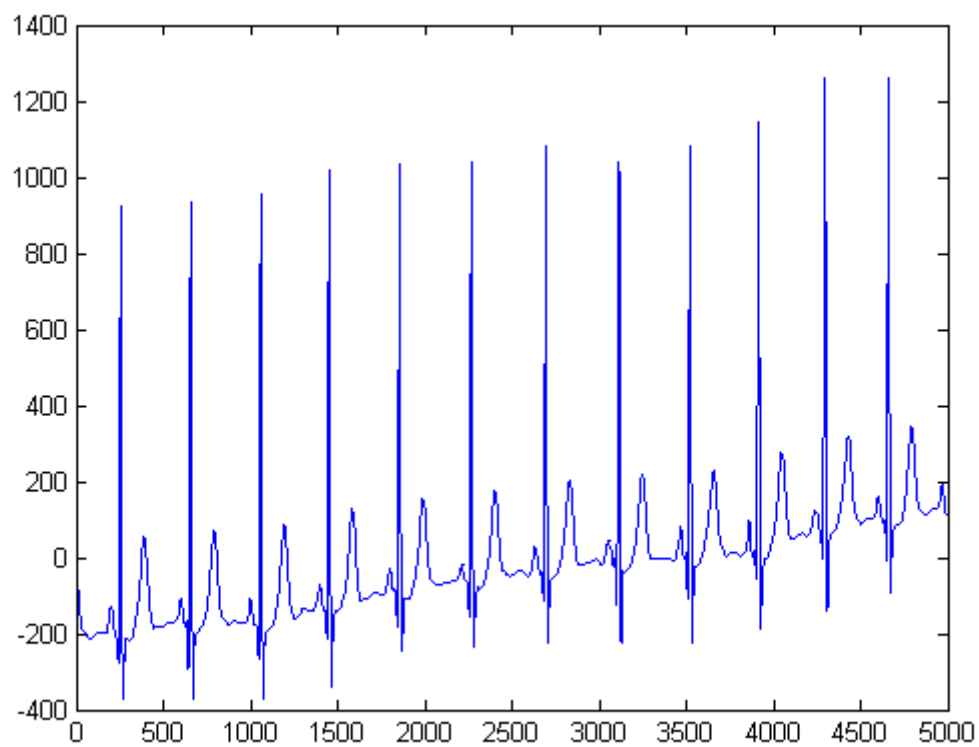
Počet pásem	Práh = 0,19 σ		Práh = 0,3 σ	
	CR	PRD [%]	CR	PRD [%]
1	2,1071	8,9939	2,2317	25,6967
2	3,7006	4,5118	4,0301	10,8438
3	5,2228	2,2779	6,2760	5,8109
4	6,3405	1,2600	8,3122	3,3224
5	6,5601	0,9863	8,9612	1,9354
6	6,5323	0,8518	8,9466	1,5699
7	6,4974	0,7358	8,8807	1,3742
8	6,4801	0,5721	8,8492	1,1092
9	6,4593	0,5516	8,8115	1,0169
10	6,4383	0,5559	8,7735	0,9598
11	6,3993	0,5850	8,7030	1,1060
12	6,3606	0,6068	8,6332	1,0334
13	6,3399	0,6146	8,5960	1,0358
14	6,3019	0,6172	8,5278	1,0384
15	6,2643	0,6172	8,4607	1,0384
16	6,2272	0,6172	8,3947	1,0384
17	6,1906	0,6172	8,3298	1,0384
18	6,1545	0,6172	8,2660	1,0384
19	6,1188	0,6172	8,2032	1,0384
20	6,0835	0,6172	8,1414	1,0384

5.3.4 Výsledný signál

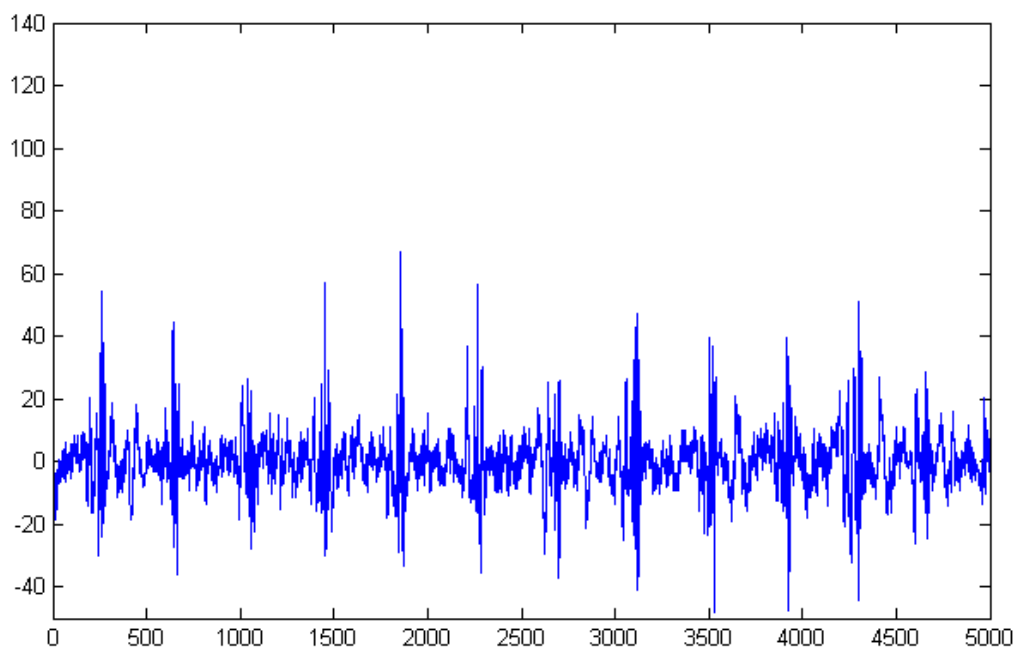
Na obrázcích 21 a 22 je srovnání původního signálu a signálu po rekonstrukci. Obrázek 23 ukazuje jejich rozdílový signál. Ten je kvůli výrazně menší amplitudě zobrazen v desetkrát větším měřítku. Jde o první svod ze souboru MO1_125_03. Během komprese byl počet pásem transformace pět a práh 0,3 násobek směrodatné odchylky. Jako mateřská vlnka byla použita vlnka bior3.1. Celkový tvar signálu se nijak výrazně nezměnil. Přesto je zřetelné, že došlo k útlumu zvláště na vyšších frekvencích. Křivka EKG je ve výsledku čistší, extrémy jsou nepatrně menší. To je dáno tím, že koeficienty nejvyšších frekvencí mají nejmenší hodnoty, a jako takové podléhají prahování. Zkreslení je nejvíce patrné na začátku kmitu Q.



Obrázek 21. Původní signál prvního svodu ze souboru MO1_125_03

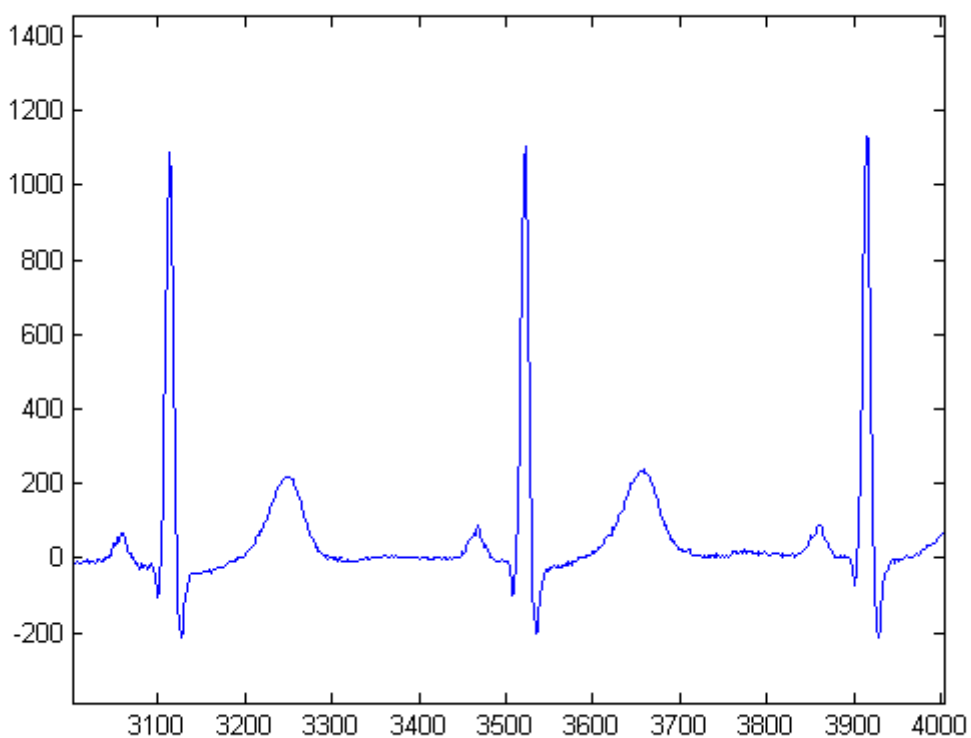


Obrázek 22. Stejný signál po rekonstrukci

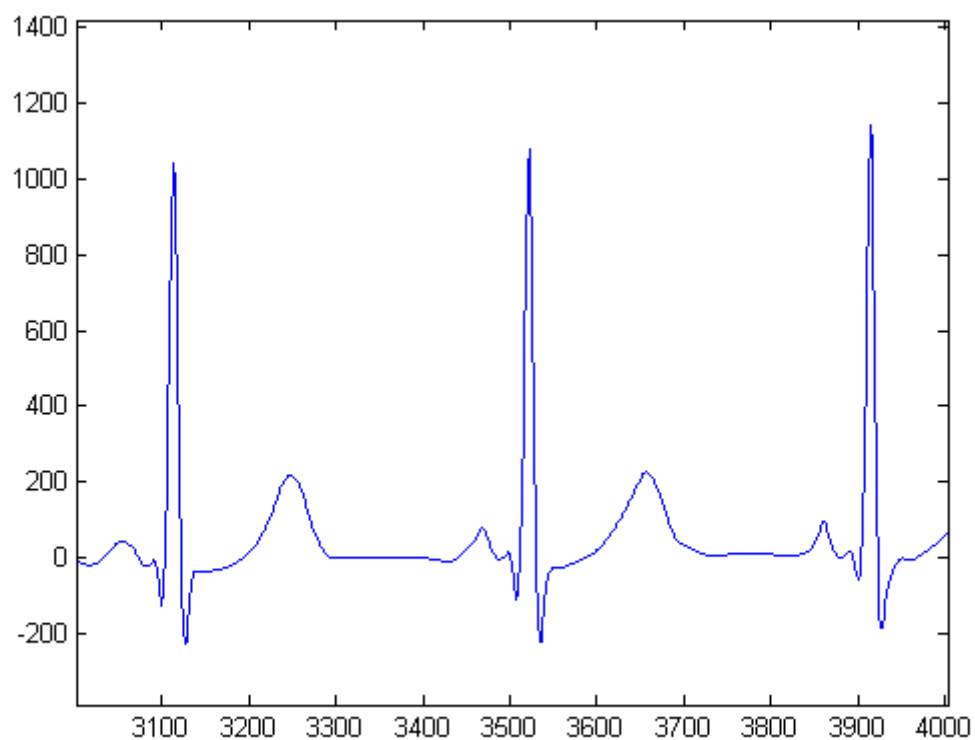


Obrázek 23. Rozdílový signál původního a rekonstruovaného průběhu (10× zvětšeno)

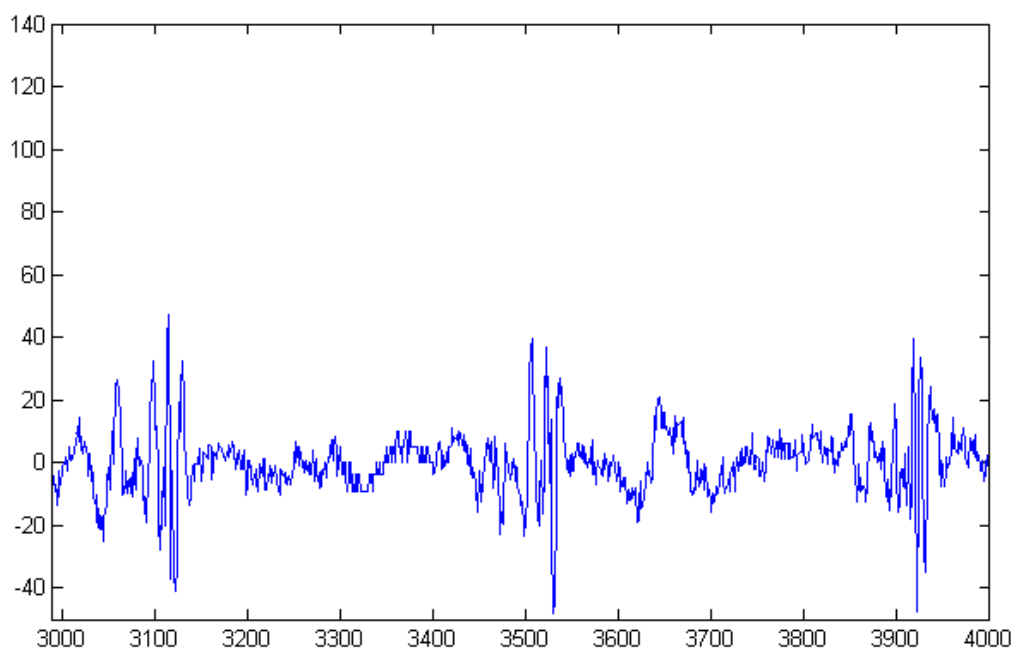
Na obrázcích 24 a 25 je detail týchž signálů. Jsou zde vidět tři periody EKG. Za těmito obrázky opět následuje další, zobrazující jejich rozdílový signál. Ten je i tentokrát desetkrát zvětšen.



Obrázek 24. Detail původního signálu



Obrázek 25. Detail rekonstruovaného signálu



Obrázek 26. Detail rozdílového signálu (10× zvětšeno)

6 Implementace Burrows-Wheelerovy transformace

Dalším implementovaným algoritmem komprese je právě Burrows-Wheelerova transformace. Princip tohoto postupu je prezentován v kapitole 2.6. V této práci jsou testovány dva způsoby její aplikace. Ty jsou popsány na následujících stránkách.

6.1 Aplikace BWT přímo na EKG signál

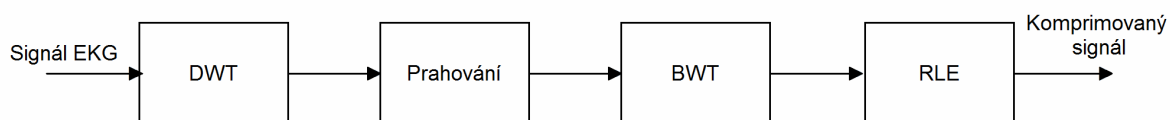
Prvním způsobem implementace BWT je její přímá aplikace na EKG signál s následnou kompresí pomocí proudového kódování. Nejlepších výsledků bylo dosaženo při transformaci každého svodu zvlášť, přesto byl výsledný kompresní poměr ještě horší, než u aplikace samotného RLE. Konkrétně bylo dosaženo CR **1,0428**.

6.2 Aplikace BWT na koeficienty DWT

Burrows-Wheelerova transformace je tentokrát aplikována na již prahované koeficienty nejvyššího pásma vlnkové transformace před RLE kódováním. Do transformace opět vstupuje každý svod signálu samostatně. Celkový postup znázorňují diagramy na obrázcích 27 a 28.

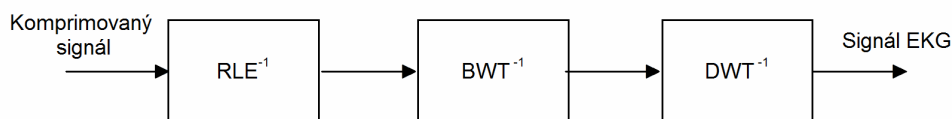
Burrows-Wheelerova transformace byla aplikována s cílem sjednotit dlouhé úseky nul v posledních pásmech vlnkové transformace. Poslední pásmo je totiž po prahování tvořeno relativně dlouhými sekvencemi nul, místy narušenými nenulovými vzorky. Tuto skutečnost ilustruje obrázek 29. Jedná se o první svod signálu MO1_001_03 rozloženého na pět pásem s prahem $0,13\sigma$.

Komprese

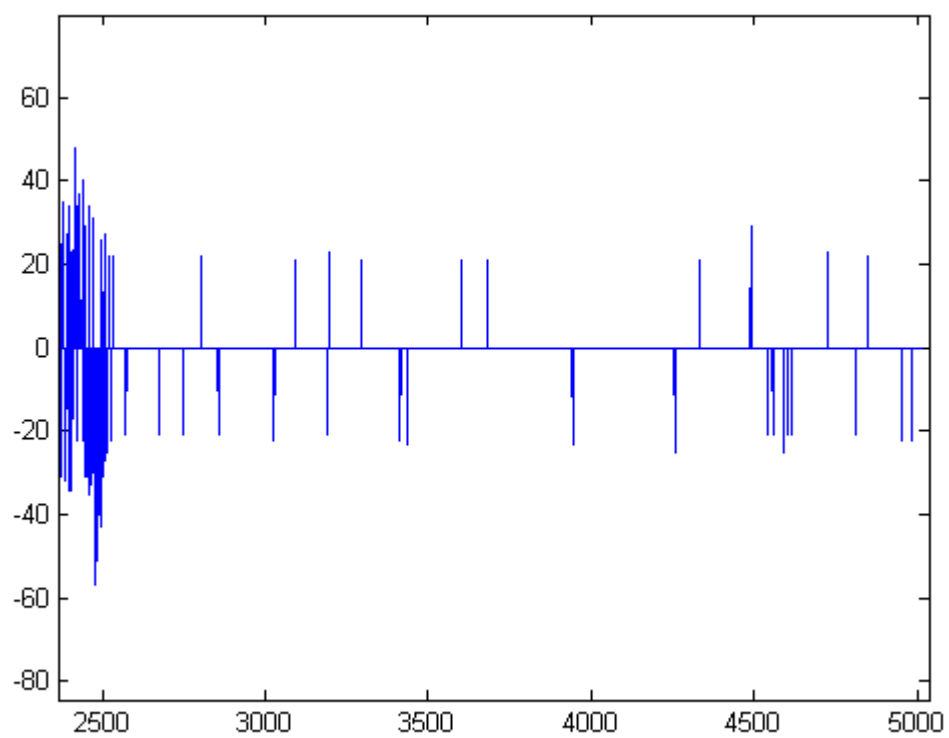


Obrázek 27. Diagram procesu komprese

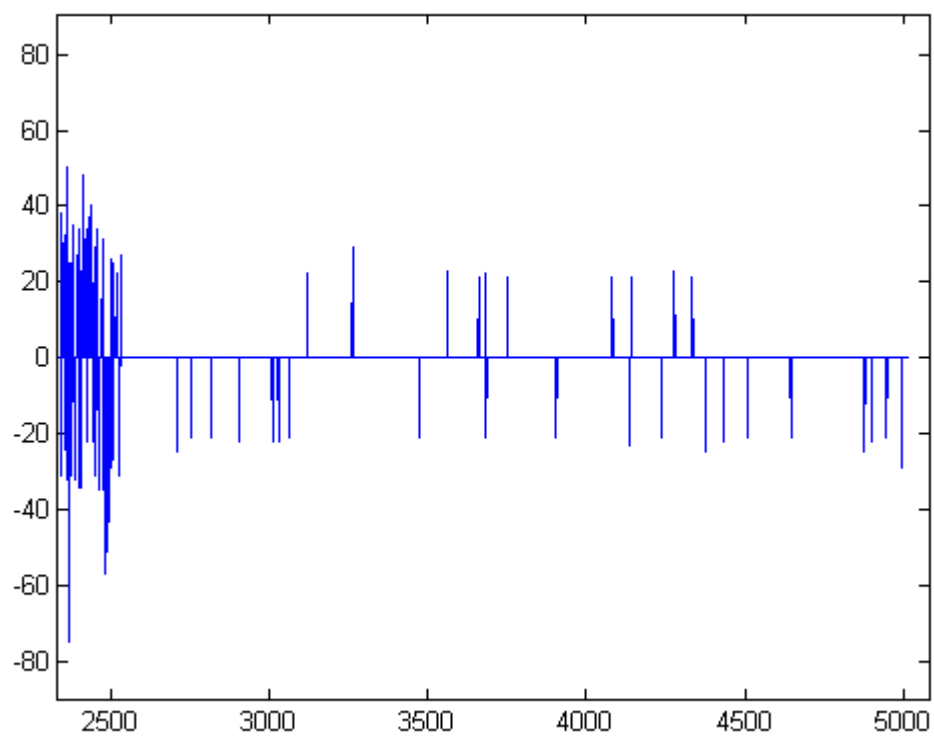
Dekomprese



Obrázek 28. Diagram procesu dekomprese



Obrázek 29. Průběh koeficientů posledního pásma vlnkové transformace



Obrázek 30. Průběh koeficientů posledního pásma DWT po aplikaci BWT

Průběh po aplikaci Burrows-Wheelerovy transformace je na obrázku 30. Uspořádání nenulových peaků se bez pochyby změnilo. Změna se ale nejeví nijak výrazně. Její vliv na proces komprese je hodnocen v další kapitole.

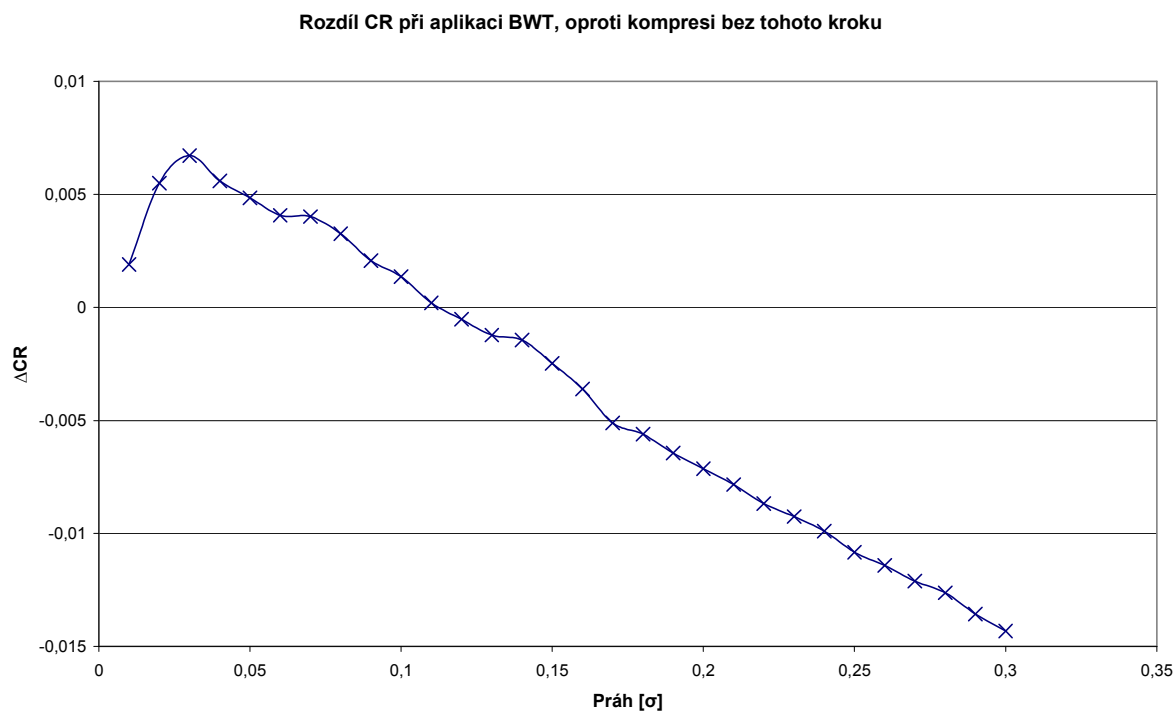
6.2.1 Zhodnocení výsledků

Podobně jako u hodnocení parametrů vlnkové transformace bylo testováno chování algoritmu při různých hodnotách prahu. Získané údaje jsou uvedeny v tabulce 3. Obrázek 31 ilustruje rozdíl kompresních poměrů se zařazenou Burrows-Wheelerovou transformací oproti kompresním poměrům dosaženým bez ní. Tento krok do komprese nevnáší žádné další zkreslení, hodnoty PRD se proto nemění a nejsou zde ani uvedeny. Rozdíl CR je určen dle následujícího vzorce:

$$\Delta CR = CR_{BWT} - CR_{DWT} . \quad (8)$$

CR_{BWT} zde reprezentuje kompresní poměr dosažený s aplikací BWT, CR_{DWT} potom kompresní poměr bez aplikace BWT.

Na obrázku 31 vidíme, že změny CR při implementaci Burrows-Wheelerovy transformace jsou mizivé. Kladné jsou navíc jen při velice nízkém prahu, kdy je kompresní poměr nezajímavý. Při vyšším prahu jsou tyto změny záporné a tudíž nežádoucí. Pro správné pochopení grafu je důležité všimnout si, kde se na ose y nachází nula.



Obrázek 31. Rozdíl CR při aplikaci BWT oproti kompresi bez BWT

Tabulka 3. Rozdíl CR při aplikaci BWT oproti kompresi bez BWT

Práh [σ]	CR_{BWT}	CR_{DWT}	ΔCR	Práh [σ]	CR_{BWT}	CR_{DWT}	ΔCR
0,01	1,1126	1,1107	0,0019	0,16	5,7921	5,7957	-0,0036
0,02	1,4604	1,4549	0,0055	0,17	6,0538	6,0589	-0,0051
0,03	1,8204	1,8137	0,0067	0,18	6,3076	6,3132	-0,0056
0,04	2,1705	2,1649	0,0056	0,19	6,5537	6,5601	-0,0064
0,05	2,5133	2,5084	0,0048	0,2	6,7942	6,8013	-0,0071
0,06	2,8473	2,8432	0,0041	0,21	7,0375	7,0454	-0,0078
0,07	3,1756	3,1716	0,0040	0,22	7,2718	7,2804	-0,0087
0,08	3,4886	3,4854	0,0033	0,23	7,5040	7,5132	-0,0093
0,09	3,7957	3,7936	0,0021	0,24	7,7250	7,7349	-0,0099
0,1	4,1015	4,1001	0,0014	0,25	7,9480	7,9588	-0,0108
0,11	4,3967	4,3965	0,0002	0,26	8,1504	8,1618	-0,0114
0,12	4,6899	4,6904	-0,0005	0,27	8,3582	8,3703	-0,0121
0,13	4,9821	4,9833	-0,0012	0,28	8,5551	8,5677	-0,0126
0,14	5,2620	5,2634	-0,0014	0,29	8,7498	8,7634	-0,0136
0,15	5,5264	5,5289	-0,0025	0,3	8,9468	8,9612	-0,0143

Kromě horšího kompresního poměru implementace BWT přinesla i výrazné zpomalení komprese. Doba jejího trvání se tak posunula o dva řády. Je to dáno tím, že oproti příkladu v kapitole 2.6.1 se zde v transformovaných sekvencích nachází vysoký počet opakujících se znaků, nul. Zatímco v příkladu stačilo pro úspěšné seřazení všech řádků matice M_1 zohledňovat pouze první dva znaky každého řádku, zde není situace tak jednoduchá. Algoritmus musí zpracovat mnohem větší množství dat. Právě řazení je přitom nejnáročnější operací celé Burrows-Wheelerovy transformace.

Kromě posledního pásma vlnkové transformace byla během testování Burrows-Wheelerova transformace aplikována i na druhé nejvyšší pásmo koeficientů DWT. Takto získaný kompresní poměr byl ale ještě horší, než u posledního pásma. Lepšího poměru se nepodařilo dosáhnout ani aplikací BWT na obě nejvyšší pásma dohromady.

7 Srovnání algoritmů

Pro jednoznačné označení algoritmů testovaných v této práci a jejich srovnání je nejdříve potřeba všechny tyto algoritmy pojmenovat. K tomu využiji postupů, které jsou v těchto algoritmech využity. Přehledně jsou tyto názvy spolu s kapitolami, kde jsem se danými algoritmy zabýval, uvedeny v tabulce 4.

Tabulka 4. Jednoznačné označení algoritmů

Algoritmus	Kapitola
RLE	4
BWT+RLE	6.1
DWT+RLE	5
DWT+BWT+RLE	6.2

Tabulka 5 obsahuje srovnání výše popsaných algoritmů s jinými používanými kompresními postupy. Některé z nich jsou navrženy přímo pro kompresi EKG (AZTEC, CORTES, SPIHT, Diohn, Hilton), jiné nejsou tak úzce zaměřené.

Tabulka 5. Porovnání s jinými algoritmy

Algoritmus	CR	PRD [%]
7ZIP	3,60	0,00
ALZ	2,50	0,00
AZTEC	10,00	28,00
CORTES	4,80	7,00
Diohn	8,00	3,90
Hilton	8,00	2,60
JAR	2,50	0,00
LZH	2,50	0,00
RAR	3,40	0,00
SPIHT	8,00	1,18
TAR	1,10	0,00
TGZ	2,60	0,00
TP	2,00	5,30
ZIP	2,50	0,00
RLE	1,05	0,00
RLE+BWT	1,04	0,00
DWT+RLE	8,96	1,94
DWT+BWT+RLE	8,94	1,94

Pro úplnost těchto údajů uvádím za jakých parametrů byly dosaženy. Zatímco algoritmy RLE a BWT+RLE nejsou závislé na žádných parametrech jiných, než parametrech vstupního signálu, u druhých dvou algoritmů je to jinak. Hodnoty pro algoritmus DWT+RLE byly dosaženy při vlnkovém rozkladu na pět pásem a prahu o velikosti $0,3 \sigma$. Stejně parametry platí i pro algoritmus DWT+BWT+RLE. Zde byla

Burrows-Wheelerova transformace uplatněna pouze na poslední pásmo koeficientů vlnkové transformace.

Část údajů uvedených v tabulce 5 je převzata z [13] [15]. Je vidět, že bezztrátové algoritmy zdaleka nedosahují takových kompresních poměrů jako ty nejlepší ztrátové. Přitom RLE a RLE+BWT dosahují tak mizivých kompresních poměrů, že se nedá mluvit o jejich praktickém využití.

Oproti tomu algoritmus DWT+RLE si vede mnohem lépe. V kompresním poměru ho předčí jen AZTEC (Amplitude Zone Time Epoch Coding), který ale dosahuje hodnot PRD o řád vyšších. Skutečným konkurentem je algoritmus SPIHT (Set Partitioning in Hierarchical Trees). Ten sice nedosahuje tak velkého kompresního poměru, ale za to má menší hodnotu PRD. Úpravou parametrů algoritmu DWT+RLE lze ovšem dosáhnout ještě lepších výsledků. Zvýšením počtu pásem vlnkové transformace se hodnota PRD už při osmi pásmech dostane pod hodnotu SPIHT a to za kompresního poměru 8,85 (viz Tabulka 2).

Srovnatelných výsledků dosahuje i algoritmus DWT+BWT+RLE, jeho kompresní poměr je jen o několik setin horší.

Závěr

Přímá aplikace proudového kódování se ukázala jako vysoce neefektivní. Přesto to neznamena, že by se při kompresi EKG měl tento postup zcela ignorovat. O tom svědčí samotný fakt, že proudové kódování je natolik rozšířené, že jej lze nalézt téměř v každém sofistikovanějším algoritmu. Své místo má i v dalších algoritmech v této práci.

Bylo prokázáno, že využití proudového kódování pro kompresi koeficientů vlnkové transformace signálu EKG umožňuje dosáhnout výrazné úspory místa na datových nosičích při minimálním zkreslení signálu. Úroveň komprese i zkreslení je přitom volitelná dle potřeb uživatele pomocí vhodného nastavení parametrů. Použitý algoritmus využívá adaptivního prahování, kdy je práh závislý na velikosti směrodatné odchylky signálu. Tato skutečnost zaručuje dobré výsledky i pro signály odlišného charakteru než testované.

Kromě komprese lze vlnkovou transformaci využít také k filtraci signálů [16]. Vzhledem k tomu, že změna průběhu signálu po rekonstrukci má charakter filtrace (odfiltrované vysokofrekvenční složky rušení), lze předpokládat, že by se obě využití transformace mohla spojit.

Implementace Burrows-Wheelerovy transformace se bohužel ukázala jako krok zpět. Lepšího kompresního poměru bylo dosaženo jen pro poměrně úzkou a nezajímavou oblast prahů. Jde o oblast nízkých prahů, kdy je pásmo podléhající BWT relativně chudé na sekvence nul. To algoritmu BWT poskytuje dostatek vzorků různých hodnot, které tak může přeskupit způsobem, kdy se stejné vzorky dostanou k sobě. Toto zlepšení kompresního poměru ale nepřevyšuje půl procenta, je proto zanedbatelné.

Při hodnotách prahu, kdy už je poslední pásmo na sekvence nul mnohem bohatší, podává Burrows-Wheelerova transformace stále horší a horší výsledky. Mezi samými nulami se zde jen výjimečně objeví nenulový vzorek. Výstup BWT je závislý na pravděpodobnostech s jakými po sobě znaky následují. Jenže zde se není čeho chytit. Po jakém znaku bude následovat nenulový vzorek? Téměř jistě po nule, jenže nul je zde příliš mnoho a jen minimum z nich skutečně předchází nenulovému vzorku. Pravděpodobnost, že jedna konkrétní nula je právě tou, po které přijde nenulový vzorek, je příliš nízká. Výsledkem je sice přeskupení nenulových vzorků, toto přeskupení ale téměř nikdy nevede ke zlepšení kompresního poměru.

Jakkoli jsou změny dosažených kompresních poměrů minimální, nárůst výpočetní náročnosti je skutečně markantní. Konkrétní doba běhu algoritmu je závislá na velikosti prahu, přitom můžeme říct, že čím větší práh, tím déle krok BWT potrvá. To je dáno

vzrůstajícím počtem stejných znaků v signálu (v našem případě nul). Seřadit abecedně sekvence téměř stejných znaků si vyžaduje kontrolu příliš mnoha z nich. V průměru se potom doba běhu algoritmu začleněním BWT prodloužila zhruba stokrát.

Burrows-Wheelerova transformace pohořela i při přímé aplikaci na EKG signál, kdy předcházela proudovému kódování. Zde se alespoň neobjevil tak velký nárůst výpočetní náročnosti. Doba běhu komprese se prodloužila jen asi desetkrát.

Testované způsoby implementace BWT se neprojevily jako příznivé, přesto nevylučují, že existují jiné postupy, které BWT dokáží využít ke skutečnému zlepšení komprese.

Seznam použité literatury

- [1] ANNADURAI, S., GEETHA, P. Efficient Secured Lossless Compression of Medical Images - Using Modified Runlength Coding for Character Representation. In *Indicon 2005 Conference*, Chennai, Indie, Dec. 2005
- [2] SHANNON, Claude E. A Mathematical Theory of Communication. *Bell System Technical Journal*, 1948, vol. 27, pp. 379-423, 623-656, July, October
- [3] SALOMON, D. *Data Compression - The Complete Reference*. London: Springer, 2007. 1092 p. 4.ed. ISBN 1-84628-602-5
- [4] WITTEN, Ian H. MOFFAT, Alistair. *Managing Gigabytes: Compressing and Indexing Documents and Images*. Second edition. San Diego: Academic Press, 1999. 519 p. ISBN: 1-55860-570-3
- [5] MÍKA, Radek. *Statistické metody komprese* [online]. [cit. 2010-12-25] Dostupné z: <<http://radekmika.cz/kompresse/statisticke-metody-kompresse.html>>
- [6] BENTLEY, Jon L., SLEATOR, Daniel D. A Locally Adaptive Data Compression Scheme. *Communications of the ACM*. 1986, Volume 29, Number 4, pp. 320-330. ISSN: 0001-0782.
- [7] DAUBECHIES, Ingrid. *Ten Lectures on Wavelets*. 9th printing. Philadelphia: SIAM, 2006. 365 pp. ISBN: 0-89871-274-2.
- [8] SOMAN, K., RAMACHANDRAN, K. I. *Insight into Wavelets: From Theory to Practice*. 5th printing. New Delhi: Prentice-Hall of India, 2004. 308 pp. ISBN: 81-203-2650-4.
- [9] KRISHNAMACHARI, Parashar. *Discrete Wavelet Transform* [online]. 2003 [cit. 2011-05-11]. Dostupné z: <<http://www.thepolygoners.com/>>
- [10] BURROWS, Michael, WHEELER, David J. A Block-sorting Lossless Data Compression Algorithm. 1994. 18 pp.
- [11] GUYTON, Arthur C.; HALL, John E. *Textbook of Medical Physiology*. 11th ed. Philadelphia: Elsevier Inc., 2006. 1116 p. ISBN 0-216-5944-6
- [12] HAMPTON, R. J.: EKG stručně, jasně, přehledně. 2. vydání. Praha: Grada Publishing a.s., 2005. 152 s. ISBN 80-247-0960-0
- [13] HORSPOOL Nigel R., WINDELS Warren J. An LZ approach to ECG compression, In: *Proc. IEEE Symp. Computer-Based Medical Systems*, 1994, pp. 71-76
- [14] SWERDLOW, Charles D., SCHSLS, Wolfgang., DIJKMAN, Barbara., et al. Detection of atrial fibrillation and flutter by a dual chamber implantable cardioverter defibrillator. *Circulation*, 2000, 101, p. 878-885.
- [15] LU, Zhitao, KIM, Youn D., PEARLMAN, William A. Wavelet Compression of ECG Signals by the Set Partitioning in Hierarchical Trees (SPIHT) Algorithm. *IEEE Transactions of Biomedical Engineering*. 2000. 15 pp.
- [16] KOZUMPLÍK, Jiří. *Vlnkové transformace a jejich využití pro filtraci signálů EKG*, Teze habilitační práce. Brno: VUT, 2005. 27 s. ISSN: 1213-418X

Seznam použitých zkratek

AZTEC	Amplitude Zone Time Epoch Coding
BWT	Burrows-Wheelerova transformace
CSE	Common Standards for Quantitative Electrocardiography
CR	Compress Rate – kompresní poměr
DWT	Discrete Wavelet Transform – diskrétní vlnková transformace
EKG	elektrokardiogram
LZ	Lempel-Ziv
LZSS	Lempel-Ziv-Storer-Szymanski
LZW	Lempel-Ziv-Welch
MTF	Move-to-Front
PRD	Percent Root-mean-square Difference
RLE	RunLength Encoding - proudové kódování
SPIHT	Set Partitioning in Hierarchical Trees
TP	Turning Point